



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

## NÁVRH ELEKTRONICKÉHO OBCHODU

E-COMMERCE DESIGN

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Michal Kolář

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Luhan, Ph.D., MSc

BRNO 2021

## Zadání bakalářské práce

Ústav: Ústav informatiky  
Student: **Michal Kolář**  
Studijní program: Systémové inženýrství a informatika  
Studijní obor: Manažerská informatika  
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**  
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

### Návrh elektronického obchodu

#### Charakteristika problematiky úkolu:

Úvod  
Cíle práce, metody a postupy zpracování  
Teoretická východiska práce  
Analýza současného stavu  
Vlastní návrhy řešení  
Závěr  
Seznam použité literatury  
Přílohy

#### Cíle, kterých má být dosaženo:

Navrhnout elektronický obchod konkrétního subjektu dle jeho specifických požadavků s využitím HTML, CSS, SQL, PHP a frameworku Laravel.

#### Základní literární prameny:

BEGG, C., R. HOLOWCZAK a T. CONOLLY. Mistrovství - Databáze: Profesionální průvodce tvorbou efektivních databází. 1. vyd. Praha: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

CAMERON, D. HTML5, JavaScript, and jQuery: 24-Hour Trainer. 1st ed. Indianapolis: John Wiley & Sons, 2015. 408 p. ISBN 978-1-119-00116-4.

HOGAN, B. P. HTML5 and CSS3: Výukový kurz webového vývoje. 1. vyd. Brno: Computer Press, 2011. 272 s. ISBN 978-80-251-3576-1.

ROUBÍČEK, L. Programujeme PHP profesionálně. 2. vyd. Praha: Computer Press, 2002. 656 s. ISBN 80-7226-310-2.

WELLING, L. a L. THOMSON. Mistrovství PHP a MySQL. 1. vyd. Brno: Computer Press, 2017. 800 s. ISBN 978-80-251-4892-1.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně dne 28.2.2021

L. S.

---

Mgr. Veronika Novotná, Ph.D.  
ředitel

---

doc. Ing. Vojtěch Bartoš, Ph.D.  
děkan

## **Abstrakt**

Tato bakalářská práce se zabývá vytvořením návrhu a realizací elektronického obchodu pro nově vzniklou společnost, zaměřenou především na servis Apple zařízení. Práce se soustředí na analýzu společnosti, požadavků a implementaci elektronického obchodu s inovativními prvky.

## **Klíčová slova**

Elektronický obchod, Laravel, framework, analýza, implementace

## **Abstract**

Bachelor's thesis deals with a creation of design and implementation of e-commerce for the newly formed company, focused primarily on Apple devices service. The work focuses on the analysis of the company, requirements and implementation of e-commerce with innovative elements.

## **Key words**

E-commerce, Laravel, framework, analysis, implementation

### **Bibliografická citace**

KOLÁŘ, Michal. *Návrh elektronického obchodu* [online]. Brno, 2021 [cit. 2021-05-14]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/135352>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jan Luhan.

### **Čestné prohlášení**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 15. května 2021

---

podpis studenta

### **Poděkování**

Děkuji tímto Ing. Janu Luhanovi, Ph.D., MSc za jeho cenné rady a konzultace při vedení mé práce. Dále bych rád poděkoval svému dobrému příteli za pomoc s technickými otázkami a v poslední řadě své trpělivé přítelkyni za veškerou podporu, kterou mi poskytla.

# Obsah

Úvod.....	10
Cíle práce, metody a způsoby zpracování .....	11
1. Teoretická východiska práce .....	12
1.1. Webové technologie.....	12
1.1.1. HTML .....	12
1.1.1.1. Historie .....	13
1.1.1.2. Struktura .....	13
1.1.1.3. HTML tagy .....	14
1.1.2. CSS .....	15
1.1.2.1. Strategie použití CSS.....	16
1.1.2.2. Vazba na HTML dokument.....	16
Prvek <link> .....	16
Prvek <style> .....	17
1.1.3. SASS.....	17
1.1.4. JavaScript.....	18
1.2. Programovací jazyky pro práci s webem .....	19
1.2.1. PHP a MySQL .....	19
1.2.1.1. Framework Laravel .....	20
1.2.1.2. phpMyAdmin .....	22
1.3. Elektronický obchod .....	23
2. Analýza současného stavu .....	24
2.2. O společnosti.....	24
2.2.1. Nabízené služby.....	24
2.2.2. Způsob oprav zařízení.....	25
2.2.3. Propagace.....	26
2.2.4. Současní zákazníci .....	26
2.1. Specifikace pro elektronický obchod .....	27
3. Vlastní návrhy řešení .....	30
3.1. Návrh elektronického obchodu .....	30
3.1.1. Relační model .....	30
3.1.2. Design elektronického obchodu.....	32
3.2. Implementační detaily .....	33



3.2.1.	Nastavení a instalace .....	33
3.2.2.	HTML a CSS .....	35
3.2.2.1.	Hlavní strana .....	37
3.2.2.2.	Košík .....	37
3.2.2.3.	Dokončení objednávky .....	38
3.2.2.4.	Servis .....	38
3.2.3.	PHP, JavaScript .....	40
3.2.3.1.	Databáze .....	40
3.2.3.2.	PHP .....	44
3.3.	Administrátor .....	50
3.4.	Uživatelské rozhraní .....	52
3.4.1.	Úvodní strana .....	52
3.4.2.	Registrace a přihlášení .....	52
3.4.3.	iPhone, Macbook, Watch, iPad .....	52
3.4.4.	Košík .....	52
3.4.5.	Dokončení objednávky .....	53
3.4.6.	Servis .....	53
3.4.7.	O nás .....	53
3.5.	Ekonomické zhodnocení .....	54
4.	Závěr .....	55
	Seznam použité literatury .....	56
	Seznam obrázků .....	58
	Seznam příloh .....	60

## Úvod

V dnešní době existuje enormní množství elektronických obchodů, které jsou specializovány na více služeb než pouhé prodeje a nákupy. Některé slouží jako rozšíření kamenné prodejny, jiné zase využívají pouze elektronické obchodování, a proto mezi nimi panuje obrovská konkurence.

Jako obyčejní uživatelé známe pouze ty nejrozšířenější společnosti s největší reklamou, u kterých věříme v určitou kvalitu v závislosti na množství odběratelů. Tyto společnosti s jejich růstem rozšiřují sortiment a vyvířejí sekundární služby, které nemusejí dosahovat předpokládaných kvalit a v elektronickém obchodě jsou tragicky vyřešeny. Tímto vytvářejí prostor na trhu pro nově vzniklé společnosti, které se specializují primárně na tyto služby a mohou se dále rozvíjet.

Má bakalářská práce se bude zabývat vytvořením návrhu elektronického obchodu s inovativním rozhraním, pro servis Apple zařízení, na základě požadavků nově vzniklé společnosti XYZ. První část bude věnována teorii, kde bude vysvětleno, jaké jazyky se využívají pro tvorbu elektronického obchodu. V další části bude analyzována samotná společnost, její vstup do odvětví, silné a slabé stránky. Poslední část nás bude provázet kompletními postupy a implementací, kde si vysvětlíme důvody využití určitých nástrojů a v čem spočívají jejich výhody.

## **Cíle práce, metody a způsoby zpracování**

Hlavním cílem mé bakalářské práce je navrhnout elektronický obchod společnosti XYZ, která se zabývá opravou Apple zařízení, dle specifických požadavků a využití HTML, CSS, PHP a frameworku Laravel.

Následně bude vytvořena analýza subjektu. Analýza je nezbytná z důvodu zjištění slabých míst elektronického obchodování konkurenčních firem z hlediska sekundárních služeb a v závislosti na těchto skutečnostech navrhnout a vytvořit inovativní řešení.

Celá implementace bude chronologicky popsána od samotného návrhu. V každé části postupu bude následovat vysvětlení, proč byla zvolena právě tato možnost a v čem spočívá její výhoda.

Cílem této práce není pouze vytvoření elektronického obchodu s moderním řešením dané služby, které by oslovilo více zákazníků, jedná se také o zjednodušení správy a intuitivnost stránek pro správce systému společnosti XYZ.

Aby byly naplněny cíle mé práce, je nutné splnit následující postup:

- Vypracovat teoretický základ jazyků, webových technologií a nástrojů, které jsou nezbytné k vysvětlení problematiky
- Analýza aktuálního stavu společnosti a popis zadaných specifikací pro elektronický obchod
- Návrh elektronického obchodu s detailem na design, implementaci, využití nástroje a uživatelské rozhraní
- Celkové zhodnocení návrhu elektronického obchodu

# 1. Teoretická východiska práce

Tato část se bude zabývat detailnějším popisem programovacích jazyků a pojmů, které budou následně využity v práci. Bez těchto jazyků by prakticky žádné elektronické obchodování nikdy nevzniklo. Každý z nich má specifické zaměření (designové nebo logické), ale ve výsledku jsou všechny provázané a úzce spolu souvisí. Spojením těchto jazyků vzniká elektronický obchod, který má hlubší význam a není jednoznačně definován, proto bude dále detailněji popsán.

## 1.1. Webové technologie

Dříve se používaly webové technologie především k procházení webového obsahu, čtení emailů, případně prohlížení novinek. Postupem času se tato technologie vypracovala natolik, že si bez ní život umíme jen stěží představit.

V posledních letech, kdy se každý snaží ušetřit co nejvíce času, se stává samozřejmostí vyvíjení a optimalizace veškerých technologií i pro mobilní telefony.

Tyto technologie se každým dnem vyvíjejí, zdokonalují a rozrůstají neskutečným tempem. Dnes už může prakticky kdokoli vytvořit webové stránky za pomoci následujících jazyků.

### 1.1.1. HTML

HTML je zkratkou pro *HyperText Markup Language*. Jedná se o značkovací jazyk, který je určen k vytváření dokumentů, obsahujících hypertextové odkazy a pokročilejší formátování. Podstatou dokumentu HTML je textový dokument ukládaný s příponou `.htm` nebo `.html`. Poskytuje elementy neboli *tagy*, pomocí nichž lze vytvářet, formátovat nebo upravovat webové stránky. Zdrojový kód HTML není nijak překládán a nešíří se dále jako samostatný spustitelný soubor. Po celou dobu návrhu, ladění a prohlížení zůstává v textové podobě. O jeho zobrazení se starají prohlížeče, jako jsou například Mozilla Firefox, Google Chrome, Opera a další. Tyto prohlížeče fungují jako *interpreter* jazyka HTML, přečtou si dokument řádek po řádku a podle zadaného kódu upravují vzhled dokumentu. Ke psaní HTML se dá využít i pouhý poznámkový blok

nebo jakýkoliv jiný textový editor. Není citlivý na velikost písma, tudíž můžeme *tagy* psát jak velkými, tak malými písmeny bez změny funkcionality. (1, s. 7)

#### **1.1.1.1. Historie**

V roce 1990 se objevila první verze HTML, 1.0 a ve druhé polovině 1993 byla představena neoficiální verze HTML+. Tato verze zahrnovala rysy formulářů, tabulek, obrázků s popiskem, ale neobsahovala formátování odstavců ani úpravy textu. HTML 2.0, se v roce 1994 stala oficiálním standardem. Byla to první verze, která měla formální specifikaci. Tabulky, možnost obtékání obrázků textem a matematické elementy se poprvé objevily v roce 1995 ve verzi HTML 3.0. Verze HTML 4.0 přidala podporu prvku `<object>`, který slouží ke vkládání obrázků a multimédií. Přibyla i podpora dalších funkcí například kaskádové styly, úpravy formulářů a tabulek, skriptování na straně klienta, internacionalizace neboli rozpoznávání jazyků se speciálními znaky, rozšíření znaků pro matematické operace. V dnešní době se aktivně využívá verze HTML 5.0, která vznikla v roce 2013 a přinesla podporu v prohlížečích Google Chrome, Opera, a Android mobilních zařízení. (2, s. 18)

#### **1.1.1.2. Struktura**

Každý dokument má přesně stanovenou základní kostru. Prohlížeče nedetekují chyby, které se vyskytnou při psaní kódu, na rozdíl od jiných programovacích jazyků. Pokud na nějakou narazí, jednoduše jí ignoruje a čte následující instrukce, proto je velice obtížné nalézt chybu, když program nefunguje správně. (2, s. 6)

- **<html>** - začátek HTML dokumentu
- **<head>** - hlavička dokumentu
- **<title>** - popis okna dokumentu, který je volitelný
- **<body>** - tělo dokumentu
- **<header>** - označení části těla dokumentu (menu)
- **<footer>** - označení částí těla dokumentu (zápatí)

```

1  <html>
2      <head>
3          <title>Laravel</title>
4      </head>
5
6      <body>
7
8          <header>
9          </header>
10
11         <footer>
12         </footer>
13
14     </body>
15 </html>
16

```

**Obrázek č. 1: Struktura HTML**  
(Zdroj: Vlastní zpracování)

### 1.1.1.3. HTML tagy

- **<main>** - využívá se pro hlavní obsah
- **<section>** - využívá se pro určité části stránek
- **<div>** - využívá se jako jednoduchý odstavec
- **<article>** - využívá se jako příspěvky a články
- **<nav>** - využívá se jako hlavní navigace stránek
- **<video>** - využívá se pro vkládání videí
- **<audio>** - využívá se pro vkládání audia
- **<br>** - využívá se jako přechod na nový řádek
- **<b>** - využívá se ke zvýraznění textu
- **<span>** - využívá se pro následné aplikování scriptů nebo stylů
- **<h1>** - využívá se pro určení velikosti textu pro nadpisy (3)

### 1.1.2. CSS

Kaskádové styly neboli *Cascading Style Sheets* je jazyk, využívaný HTML pro formátování dokumentu. Popisuje podobu stránek a styl jednotlivých prvků, a přitom nijak neovlivňuje obsah dokumentů. Pokud odstraníme definice stylu nebo je nepoužijeme, dokument se vrátí do původní podoby. Před nástupem CSS se HTML dokumenty formátovaly prostřednictvím mnoha rozšíření HTML, které byly postupem času nedostačující. CSS k formátování prostředků představuje vhodný doplněk, ačkoli jeho význam je širší. (5)

Využití v praxi:

- Širší formátovací možnosti
- Snadná tvorba a údržba konzistentního stylu
- Oddělení struktury a stylu
- Vyšší přístupnost dokumentů
- Dynamická práce se styly
- Formátování XML dokumentů

(4, s. 14)

CSS nabízí mnohem rozsáhlejší formátovací možnosti než HTML.

	HTML	CSS
<hr>	Lze určit horizontální zarovnání, sílu čáry a styl čáry (3D nebo plochá). Nelze ovlivnit barvu.	Pomocí vlastnosti <i>border</i> lze vytvořit až 8 stylů čáry o libovolné síle a barvě. Kromě horizontálních lze tvořit i vertikální čáry.
<ol>	Lze určit jeden ze tří možných stylů odrážky.	Pomocí vlastnosti <i>list-style-image</i> lze pro odrážky použít libovolný obrázek.
<font>	Lze určit velikost písma v pevně daných stupních, rodinu písma a barvu písma.	Velikost písma lze nastavit vlastností <i>font-size</i> v mnoha různých jednotkách, nebo relativně k jinému rozměru. Spolu s barvou písma lze určit i barvu pozadí a zajistit tak čitelnost textu za všech okolností.
<b>	Nastaví tučné písmo.	Vlastností <i>font-weight</i> lze sílu písma nastavit až v devíti stupních.

Obrázek č. 2: Rozdíly CSS a HTML  
(Zdroj: 4, s. 14, 15)

### 1.1.2.1. Strategie použití CSS

Kaskádové styly lze použít různými způsoby a v různém rozsahu.

*„Nejčastěji se však v praxi využívají tři strategie použití:*

- *Obohacení formátovacích možností*
- *Veškeré formátování s výjimkou rozmístění (layoutu) prvků*
- *Formátování jen pomocí CSS, včetně rozmístění prvků*

*Žádnou z těchto strategií nelze automaticky označit za lepší či nejlepší. Záleží na podmínkách konkrétního projektu a úsudku webdesignéra, kterou ze strategií zvolí a proč.“* (4, s. 17)

### 1.1.2.2. Vazba na HTML dokument

Aby se vytvořený CSS kód projevil v HTML dokumentu, musí být nějakým způsobem připojený. K tomu využíváme základní metody.

#### **Prvek <link>**

Využívá se pro připojení předpisu uloženého v externím souboru. Jedná se o nejpoužívanější způsob připojení. Hlavní výhodou tohoto připojení je práce s jediným souborem se stylovým předpisem, což zjednodušuje správu a jednotnou grafickou úpravu dokumentu. (4, s. 23)

Prvek se vkládá zpravidla do hlavičky a má následující atributy:

- **href** – slouží k určení cesty externího souboru se stylovým předpisem
- **type** – slouží k určení typu obsahu odkazovaného souboru (text/css)
- **media** – slouží k určení média
- **title** – slouží k přiřazení textového titulu
- **rel** – slouží k určení typu vazby dokumentu (*stylesheet, alternate stylesheet*)



## Prvek <style>

Tento prvek může být vložen přímo do HTML dokumentu a stejně jako <link> se vkládá do hlavičky dokumentu. Disponuje třemi stejnými atributy jako <link> a to:

- **type**
- **media**
- **title**

Tento styl je velice nepraktický a neodděluje styly od dokumentu tak zřetelně. Nevýhodou tohoto prvku je opakované načítání s každou novou stránkou. Může dojít k nežádoucí nekonzistenci stylu kvůli změně, která se musí provést v každém dokumentu zvlášť.

### 1.1.3. SASS

SASS (*Syntactically Awesome StyleSheets*) je jazyk určený pro stylizaci, který se překládá do již zmíněného CSS a slouží jako jeho rozšíření. Jde o svobodný software, volně šířený pod licencí MIT. Syntax využívá soubory s koncovkou `.scss`, ve kterých lze využívat mnoha zjednodušení a funkcí, které zajišťují přehlednost a větší porozumění při implementaci. (6)

Základní rozdíly jsou viditelné na obrázku (viz. Obrázek č. 3). Hlavní výhodou je postupné zanořování specifických sekcí, kde není potřeba opakovat název nadřazené sekce, ale stačí, když v rámci této sekce vypíšeme potřebnou podsekcí se složenými závorkami.

<pre> .watch-section{   display: grid;   grid-template-columns: 1fr 3fr;   margin: 100px auto 82px;   padding-top: 2%;    .sidebar li.active {     font-weight: 500;   }    .sidebar{     padding-top: 45%;     a{       color: ■#535353;     }      li{       font-size:medium;     }   }    .products {     padding-top: 5%;     display: grid;     grid-template-columns: 1fr 1fr 1fr;     grid-gap: 60px;   } } </pre>	<pre> .watch-section {   display: grid;   grid-template-columns: 1fr 3fr;   margin: 100px auto 82px;   padding-top: 2%; } .watch-section .sidebar li.active {   font-weight: 500; } .watch-section .sidebar {   padding-top: 45%; } .watch-section .sidebar a {   color: ■#535353; } .watch-section .sidebar li {   font-size: medium; } .watch-section .products {   padding-top: 5%;   display: grid;   grid-template-columns: 1fr 1fr 1fr;   grid-gap: 60px; } </pre>
--	--

**Obrázek č. 3: Rozdíly SASS (vlevo) a CSS (vpravo)**

(Zdroj: Vlastní zpracování)

#### 1.1.4. JavaScript

JavaScript je skriptovací jazyk, který se používá pro práci s HTML. Skripty se zapisují přímo do HTML kódu stránky a jsou interpretovány až na klientském počítači. Díky tomu není zapotřebí žádných speciálních nástrojů nebo překladačů a zdrojové kódy jsou všem přístupné. (7, s. 13)

Jedná se o jazyk objektový, což umožňuje práci s objekty. Právě do těchto objektů je zapouzdřeno okno prohlížeče, obsah stránky a další prvky. Syntaxe vychází z programovacího jazyka C++, a proto záleží na velkých a malých písmenech. Je to beztypový jazyk, tudíž mu při deklaraci proměnné nespecifikujeme, jakého je typu.

V dnešní době se nevyužívá pouze na webových stránkách, ale díky své oblíbenosti i ve spoustě aplikací. (9)

## 1.2. Programovací jazyky pro práci s webem

Existuje mnoho programovacích jazyků pro práci s webem, ale tato práce se bude zabývat pouze těmi nejvhodnějšími a nejznámějšími pro tvorbu elektronického obchodu.

### 1.2.1. PHP a MySQL

PHP (*Hypertext Preprocessor*) je dynamický skriptovací jazyk na straně serveru, který byl navržen speciálně pro web. Datové typy nejsou vázané na proměnnou, ale na hodnotu proměnné. Je libovolně přenositelný mezi operačními systémy. (12, s. 6)

*„Do stránky HTML můžete vkládat kód jazyka PHP, jenž se provede při každé návštěvě této stránky. Váš kód jazyka PHP je interpretován webovým serverem, přičemž generuje dokument HTML nebo jiný výstup, který uvidí návštěvník.“* (8, s. 30)

Jako projekt s otevřeným zdrojovým kódem k němu máme volný přístup a můžeme jej používat, měnit nebo dále distribuovat.

Poskytuje velkou škálu frameworků, které zjednodušují a urychlují vývoj:

- **Laravel**
- **Symfony**
- **CodeIgniter**
- **Phalcon**
- **Zend**

PHP umí přistupovat k velkému spektru databázových systémů, například MySQL, SQLite, PostgreSQL a další. Velice často se využívá v kombinaci s webovým serverem *Apache*.

MySQL je velice rychlý systém pro správu relačních databází. Databáze umožňuje efektivně ukládat, vyhledávat, načítat a řadit data. MySQL server řídí veškeré přístupy k našim datům. Zajišťuje přístup velkého množství uživatelů v reálném čase, nabízí rychlý přístup a dohlíží na to, aby přístup dostali pouze oprávnění uživatelé. Pro práci využívá standardní dotazovací jazyk *SQL*. Je celosvětově nejoblíbenější databází s otevřeným zdrojovým kódem. (8, s. 31)

#### 1.2.1.1. Framework Laravel

Laravel je framework webových aplikací s expresivní a elegantní syntaxí. Autorem je softwarový inženýr Taylor Otwell, který pracoval na tomto frameworku od roku 2011. Tento framework poskytuje strukturu a výchozí bod pro vytváření aplikací. Laravel nabízí úžasný vývojářský zážitek a zároveň poskytuje výkonné funkce, jako je důkladné vkládání závislostí, expresivní vrstva abstrakce databáze, fronty a naplánované úlohy testování jednotek, integrace a další. Ať už je uživatel v PHP nebo webových frameworkcích nováčkem nebo má dlouholeté zkušenosti, Laravel je framework, který s ním může růst. Je vyladěn pro vytváření profesionálních webových aplikací a je připraven zvládnout podnikové pracovní zátěže. (10)

Díky povaze PHP přizpůsobené škálování a integrované podpoře Laravelu, pro rychlé distribuované systémy mezi paměti, jako je Redis, je horizontální škálování s Laravelem velice jednoduché. Ve skutečnosti byly aplikace Laravel snadno škálovatelné, aby zvládly stovky miliónů požadavků měsíčně. (10)

Laravel kombinuje nejlepší balíčky v ekosystému PHP a nabízí nejrobustnější dostupný framework pro vývojáře. Do rámce navíc přispěly tisíce talentovaných vývojářů z celého světa. Protože se jedná o svobodný software, který je distribuovaný pod licencí MIT, může jej používat, modifikovat, distribuovat, publikovat prakticky každý. (10)

#### Šablony

Webové frameworky obsahují nástroje pro práci se šablonami. Tyto nástroje mají za úkol oddělit HTML od kódu programovacího jazyka. Laravel využívá šablonovací systém *Blade*.

*„Blade je jednoduchý, ale výkonný šablonový systém, který je součástí Laravelu. Na rozdíl od některých šablonových modulů PHP vás Blade neomezuje v používání prostého kódu PHP ve vašich šablonách. Ve skutečnosti jsou všechny šablony Blade kompilovány do prostého kódu PHP a ukládány do mezipaměti, dokud nejsou upraveny, což znamená, že Blade přidá vaši aplikaci v podstatě nulovou režii. Soubory šablon blade používají příponu souboru `.blade.php` a jsou obvykle uloženy v adresáři `resources / views`.“* (10)

## Modely

Laravel obsahuje *Eloquent*, Objektově-Relační Mapovač (ORM), který usnadňuje interakci s databází. Při použití *Eloquent* má každá databázová tabulka odpovídající model, který se používá k interakci s touto tabulkou. Kromě načítání záznamů z databázové tabulky, vám modely *Eloquent* umožňují také vkládat, aktualizovat a mazat záznamy z tabulky. (10)

## Pohledy

Pohledy neboli *Views*, reprezentují uživatelské rozhraní aplikace. Není praktické vracet celé řetězce dokumentů HTML přímo z tras a řadičů. Pohledy poskytují pohodlný způsob, jak umístit veškerý HTML obsah do samostatných souborů. Pohledy oddělují logiku aplikace od logiky prezentace a jsou uloženy v adresáři `resources/views`. (10)

```
<!-- View stored in resources/views/greeting.blade.php -->

<html>
  <body>
    <h1>Hello, {{ $name }}</h1>
  </body>
</html>
```

Obrázek č. 4: Pohled  
(Zdroj: 10)

## Řadiče

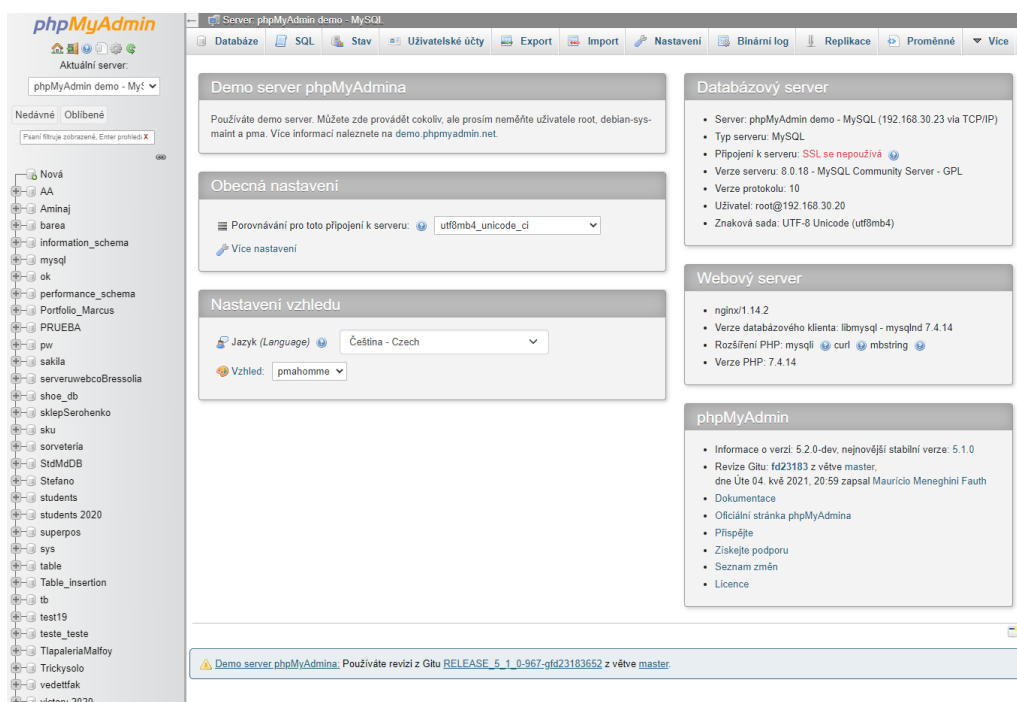
Řadiče neboli *Controllers*, pomáhají uspořádat veškerou logiku zpracování požadavků pomocí tříd řadičů. Ty mohou seskupit související logiku zpracování požadavků do jedné třídy. Například třída *UserController* může zpracovávat všechny příchozí požadavky související s uživateli, včetně zobrazování, vytváření, aktualizace a mazání uživatelů. (10)

### 1.2.1.2. phpMyAdmin

phpMyAdmin je webová aplikace napsaná v PHP programovacím jazyce, která slouží k administraci MySQL přes webové stránky. Podporuje širokou škálu operací MySQL. Často používané operace.

- správa databází
- správa tabulek
- správa sloupců
- správa relací
- správa indexů
- správa uživatelů
- správa oprávnění

Lze provádět prostřednictvím uživatelského rozhraní, přičemž je stále možnost přímo provádět jakýkoli příkaz SQL. (11)



Obrázek č. 5: phpMyAdmin rozhraní  
(Zdroj: 11)

### 1.3. Elektronický obchod

Ekonomika a podnikání se v dnešní době velmi rychle mění. Využívají se nejmodernější technologie, čímž vzniká mnoho nových příležitostí. V tomto období se začínají projevovat vlivy těchto technologií na v oblasti podnikání a obchodu, a to jak na úrovni podnikové, tak mezipodnikové. Elektronické obchody probíhají v prostředí veřejného internetu. Díky pokroku se stále zdokonalují přenosové technologie, výpočetní systémy a jejich aplikační vybava, což vytváří podmínky pro realizaci stále dokonalejších projektů elektronických obchodů různých forem a zaměření. (13, s. 48)

*„Výkonnost a konkurenceschopnost podniku je dnes dána zejména tím, jak rychle je podnik schopen reagovat na měnící se požadavky zákazníků, jak dlouhá je dodací lhůta, na jaké úrovni je schopnost komunikace a získávání, resp. předávání informací. Dochází k růstu konkurence v důsledku pronikání progresivních firem do vzdálených teorií a jejich trhů, současně také dochází ke splývání dosud oddělených odvětví, jako jsou telekomunikace, energetika, výpočetní technika, masmédia, obchod“* (13, s. 48)

Elektronické obchodování představuje pouze jednu z podob E-podnikání. Elektronický obchod se zaměřuje na nákup a prodej zboží a služeb. Zajišťuje veškeré obchodní aktivity prostřednictvím nejrozličnějších informačních technologií. Jednoznačná definice elektronického obchodu zatím není v literatuře definována. Jedna z definic říká, že elektronický obchod je jakákoliv forma obchodování prováděna elektronicky pomocí telekomunikačních sítí. Jiná zase chápe elektronické obchodování jako výměnu informací pomocí elektronického média za účelem uzavření elektronického obchodu. (13, s. 49)

*„Elektronický obchod je výhradně a jedině obchodem, který je realizován prostřednictvím počítačů a elektronické sítě. Nejznámější, byť ne jediná síť, je označována jako internet. Proto se elektronické obchody často označují jako obchody internetové. Prostřednictvím internetu lze tedy obchodovat, tj. vybrat, objednat, zaplatit zboží a služby a zajistit jeho dodání spolu s příslušnou dokumentací.“* (13, s. 50)

Na základě těchto vyjádření, elektronický obchod je obchodní transakcí, jejíž všechny fáze jsou prováděny prostřednictvím elektronických prostředků. (13, s. 50)

## **2. Analýza současného stavu**

Tato část se bude věnovat analýze společnosti XYZ, pomocí níž jsme schopni identifikovat její nedostatky a podle toho vytvořit ideální elektronické obchodování pro následný rozvoj. V tuto chvíli se společnost XYZ věnuje pouze opravám Apple zařízení, ale chce své služby rozšířit i o nákup elektroniky.

### **2.2. O společnosti**

Společnost XYZ se nachází v Jihomoravském kraji, přímo v centru Brna. V samotném centru města se nachází velké množství konkurenčních podniků, které mají kamennou prodejnu nebo servis a elektronické obchodování, proto je třeba vytvořit analýzu a zjistit silné a slabé stránky, případně nalézt prostor pro zlepšení a inovaci.

#### **2.2.1. Nabízené služby**

V současné době se společnost věnuje pouze opravám Apple zařízení.

- **Apple Watch**
- **iPad**
- **iPhone**
- **Macbook**

Tyto služby ale začínají být nedostačující a pouhé opravy nepřilákají tolik zákazníků, proto bude společnost rozšiřovat služby o prodej těchto zařízení v novém elektronickém obchodě. Elektronický obchod každopádně neznamená okamžitý úspěch. Podle statistiky existuje v České republice 41 251 elektronických obchodů a z toho 5% je zaměřených na prodej s elektronikou, což činí přes dva tisíce elektronických obchodů s prodejem elektroniky (viz. Obrázek č. 7). Z tohoto důvodu musí společnost přijít se způsobem, jak se odlišit.





Počet e-shopů

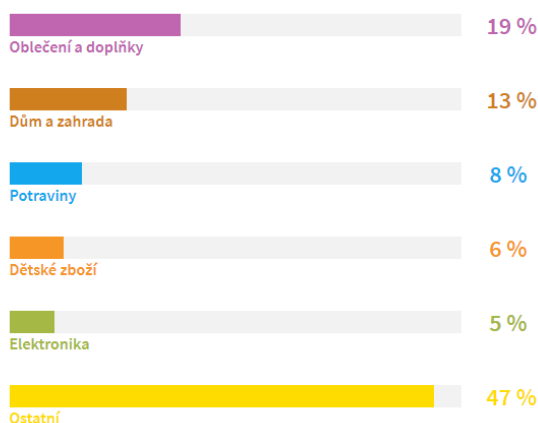
**41 251**

Do konce roku: **45 961**

Meziroční změna: **+8 %**

Nejčastější sortiment e-shopů

Dnes už v e-shopech nakoupíte skoro všechno. Který sortiment ale převládá?



**Obrázek č. 6: Počet elektronických obchodů, zaměřených na prodej elektroniky v ČR**

(Zdroj: <https://www.ceska-ecommerce.cz/#reseni>)

### 2.2.2. Způsob oprav zařízení

Opravy fungují pouze na bázi osobního setkání ve společnosti nebo telefonickým spojením. Zákazník je vždy tázán na závady a důvody jejich vzniku. Následně probíhá diagnostika a v závislosti na telefonickém schválení nebo zamítnutí cenové nabídky se zařízení opraví nebo pouze zaplatí částka za diagnostiku.

### **2.2.3. Propagace**

Společnost XYZ nevyužívá žádnou placenou internetovou propagaci v žádné formě. Spoléhá se pouze na tištěné letáky. Povědomí o této společnosti se šíří především pomocí stálých zákazníků. Jedná se o lidi spokojené se servisem natolik, že neváhají doporučit společnost rodině a známým. Pro budoucí prosperitu společnosti XYZ by bylo určitě vhodné investovat do placené internetové propagace, která by přivedla mnoho nových zákazníků.

### **2.2.4. Současní zákazníci**

Současní zákazníci společnosti XYZ se dělí do tří skupin. První skupina jsou lidé, kteří mají zastaralé, rozbité zařízení, jejichž oprava by u konkurenčních společností stála větší částku, než je aktuální hodnota zařízení.

Do druhé skupiny spadají ti, jejichž zařízení potřebuje pouze základní nebo preventivní opravu, a proto hledají společnost s nejnižší cenou.

Třetí skupina zahrnuje lidi, kteří vlastní zařízení nejnovější řady na trhu, na jejichž poškození se u oficiálních prodejců nevztahuje záruka a placená oprava se pohybuje ve výši pořizovací ceny samotného zařízení.

## **2.1. Specifikace pro elektronický obchod**

Pro začátek je nejdůležitější správně analyzovat všechny požadavky na elektronický obchod. K analýze nám poslouží veškeré informace, které nám byly poskytnuty majiteli společnosti XYZ. Na základě sdělených požadavků se nesnažíme vytvořit produkt založený na přesných požadavcích majitelů. Snažíme se vytvořit produkt, který si majitelé mysleli, že chtějí od samotného začátku.

### **Požadavky na elektronický obchod**

- Velice jednoduchý design s produkty na úvodní straně
- Žádné reklamy a křiklavé barvy
- Záložky pro každý nabízený produkt zvlášť
- Možnost přihlášení a registrace uživatelů
- Funkční nákup s košíkem, dokončení objednávky a platbou
- Jednoduchá správa databáze
- Moderní a inovativní řešení servisu zařízení, intuitivní pro uživatele s jednoduchou správou pro administrátora

## **Design a úvodní strana**

Společnost XYZ žádala o jednoduchý design stránek, kde bude menu s přehlednou navigací a logickým tříděním kategorií tlačítek. Na úvodní straně by měla být variace nabízených produktů s odkazy na podrobnosti. Konec stránky by mělo tvořit zápatí s informacemi o obchodu a jeho službách.

## **Reklamy a barvy**

Hlavní požadavek je, vyhnout se křiklavým barvám. Do těchto barev spadá červená, žlutá, fosforově zelená a podobné odstíny jmenovaných. Barvy by měly být příjemné pro oči, nejlépe s přechodem a rozdílné mezi menu a obsahem.

V souvislosti s barvami byly zamítnuty jakékoliv variace reklam, které se běžně na elektronických obchodech vyskytují. Reklamy by podle společnosti XYZ mohly budit nedůvěryhodný dojem stránek a odradit potenciální zákazníky.

## **Kategorie v menu**

Mít veškeré produkty s kategoriemi a podkategoriemi na jedné stránce se nejeví společnosti přehledné. Proto chtějí pro každou kategorii vytvořit v menu tlačítko, které jasně určuje, o jakou kategorii se jedná a třídění podle modelů se bude provádět až na individuálních stránkách.

## **Přihlášení uživatelů**

Každý návštěvník by měl v menu nalézt tlačítko pro přihlášení a tlačítko pro registraci. Při registraci musí být vstupní informace validovány a následně odesílány do databáze. Přihlášený uživatel by měl být schopný, na rozdíl od anonymního, dokončit objednávku a pokračovat k platbě

## **Košík, dokončení objednávky a platba**

Uživatel musí mít možnost přidávat libovolné zboží do košíku, měnit jeho množství a v závislosti na obsahu se musí měnit cena. Požadované ceny jsou tři, celková cena bez DPH, přepočet DPH a celková cena s DPH. Košík musí dále obsahovat dvě tlačítka, jedno k možnosti pokračování v nákupu a druhé k dokončení objednávky a zaplacení.

Při kliknutí na tlačítko dokončení objednávky bude uživatel přesměrován na její dokončení a platbu. Podle požadavků by bylo vhodné, kdyby některé osobní informace o uživateli byly automaticky převedeny do požadovaných informací na této stránce.

Společnost XYZ nemá možnost využít moderní platební brány kvůli zamítnutí velkými bankami, které se zajímají o obrát společnosti a následně schvalují tuto možnost v elektronických obchodech. Z tohoto důvodu byl přidán důležitý požadavek na vytvoření platební metody, která by byla veřejně dostupná a současně by platby monitorovala.

## **Správa databáze**

Databáze musí být přístupná administrátorovi, který bude moci upravovat, přidávat a mazat položky. Bylo by vhodné, kdyby prostředí bylo intuitivní a implementované pomocí nástroje, aby se zamezilo dlouhým odezvám nahrávání souborů.

## **Servis zařízení**

Pro anonymní uživatele bude zobrazený pouze tabulkový výpis cen a služeb jednotlivých zařízení. Přihlášený uživatel, vedený se servisovaným zařízením v databázi, musí mít přístup k informacím svého probíhajícímu servisu, který si zobrazí kliknutím na tlačítko. Po kliknutí by se měl objevit výpis těchto informací s aktuálním stavem, ve kterém se servis nachází. Pokud bude stav ve fázi dotazu platby na uživatele, objeví se návrh ceny a uživatel bude mít na výběr její potvrzení nebo zamítnutí. V obou případech bude zpřístupněna platba s potvrzením platby a v závislosti na zvolené možnosti, se bude, po potvrzení platby, odvíjet i cena za diagnostiku nebo za kompletní opravu.

### **3. Vlastní návrhy řešení**

V následující části bude detailně popsána návrhová a implementační část elektronického obchodu, společně s představením uživatelského a admin rozhraní. Tvorba elektronického obchodu bude následně ekonomicky zhodnocena v závislosti na požadavcích a výsledcích elektronického obchodu.

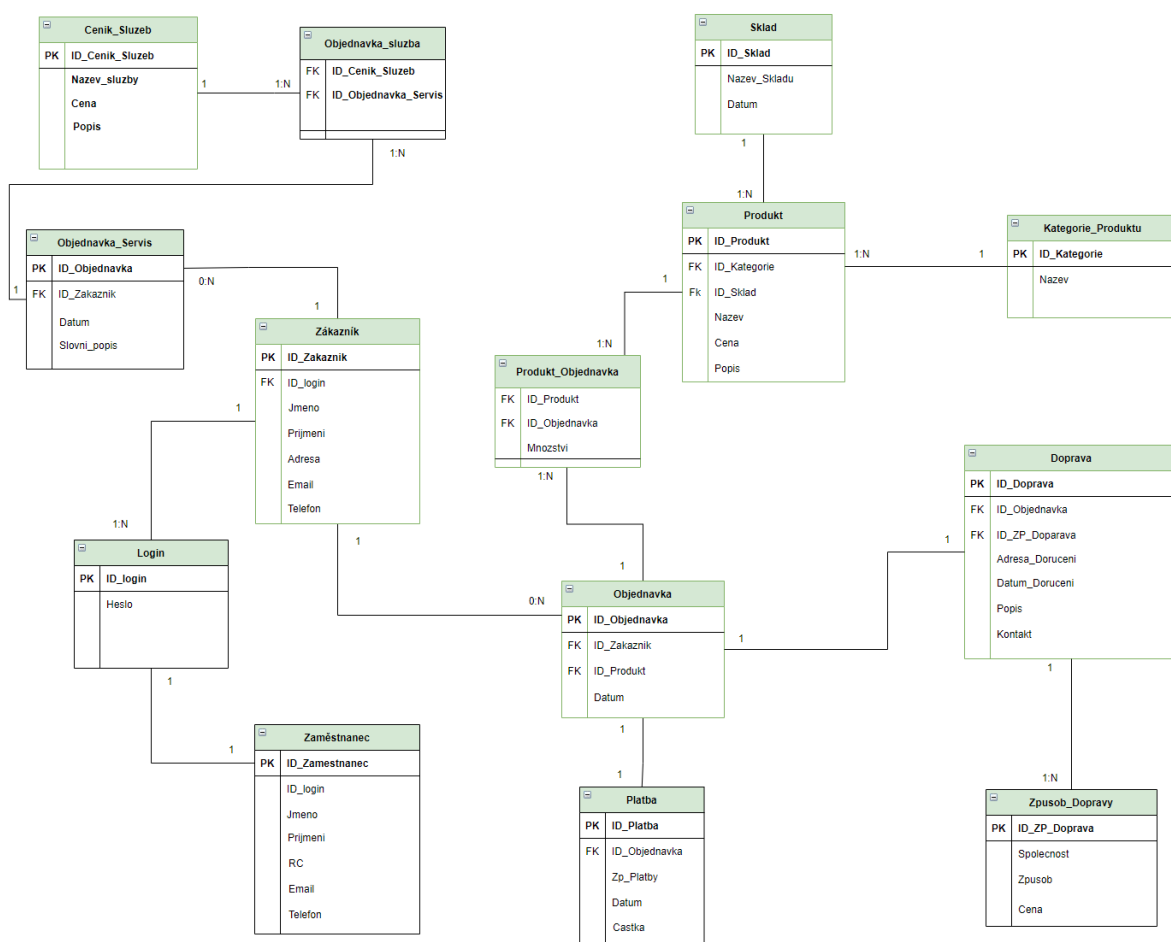
#### **3.1. Návrh elektronického obchodu**

Před samotnou implementací je třeba si uvědomit, o jaký projekt se jedná a jak postupovat při jeho realizaci. Proto je důležité si vše pečlivě připravit a rozvrhnout, co musí projekt obsahovat, jak rozmanitá bude databáze elektronického obchodu a jaké vazby mezi jednotlivými tabulkami použít.

##### **3.1.1. Relační model**

Tento model slouží pro konceptuální znázornění dat. Používá se pro analýzu požadavků v první fázi návrhu systémů. Pomocí této metody se vytvářejí Entity-Relationship Diagramy, dále jen ERD. (15)

V mém případě jsem vytvořil počáteční ERD (viz. Obrázek č. 9). Jelikož je to prvotní fáze, nikdy není tento diagram perfektní a v průběhu implementace se může měnit, případně kompletně přepracovat, pokud není vhodný.



**Obrázek č. 7: Návrh ERD elektronického obchodu**

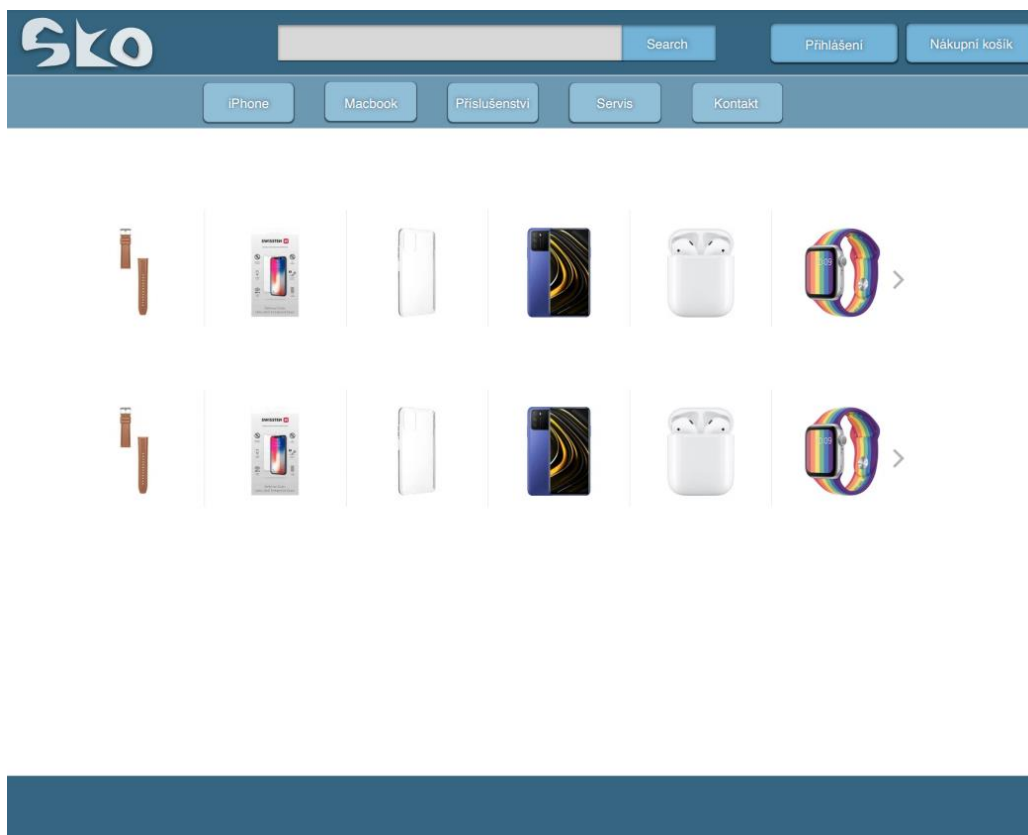
(Zdroj: Vlastní zpracování)

Zaměstnanec na obrázku je brán jako administrátor a jelikož má přístup téměř ke všem operacím v diagramu, nevytvořil jsem, z estetického důvodu, vazby k této tabulce. Na dopravu a její možnosti taktéž nebyly vytvořeny požadavky, ale bylo vhodné tyto možnosti zahrnout alespoň do návrhu v případě budoucího rozšíření. Základní logika modelu je vytvoření účtu uživatele, a následné dvě možnosti:

- **Vytvoření objednávky** – Uživatel si vybere libovolný produkt z databáze, vyplní požadované údaje, které se v databázi budou ukládat i s objednávkou, a přejde k zaplacení produktů, kde dokončí nákup.
- **Vytvoření servisu** – Uživatel na základě požadavku vytvoří servis, kde admin zpracuje obdržena data a zadá je do databáze s ohledem na cenu a identifikovanou závadu. Tímto se uživateli zpřístupní zadaná data z databáze.

### 3.1.2. Design elektronického obchodu

Nedílnou součástí návrhu je vytvoření designu stránek. Tento krok pomůže utřídit si myšlenky ohledně volby správných barev, rozvržení tlačítek v menu a celkového obsahu na každé stránce.



**Obrázek č. 8: Návrh designu hlavní strany**

(Zdroj: Vlastní zpracování)

V návrhu jsem zvolil jednoduchý a přehledný design s variacemi modré barvy, které vzbuzují pocit klidu a pohody (viz. Obrázek č. 10). Se stejnou logikou jsem vytvořil design ostatních stránek elektronického obchodu, kde se prakticky mění pouze obsah, nikoliv menu a zápatí.

V této fázi vše slouží pouze jako předloha, tudíž tlačítka, logo a veškeré informace v záhlaví a obsahu se mohou změnit.



## 3.2. Implementační detaily

Pro implementaci elektronického obchodu jsem zvolil framework Laravel, který mi oproti čistému PHP značně usnadnil práci, díky svým funkcím, které popisují v teoretické části.

### 3.2.1. Nastavení a instalace

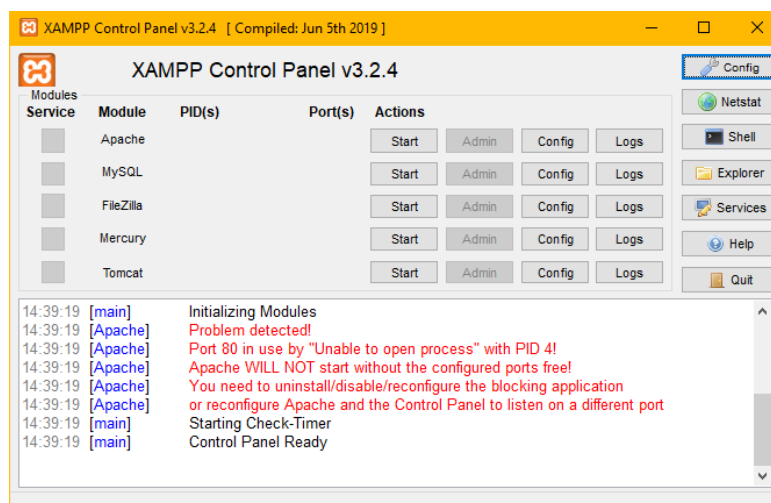
Protože jsem pracoval s verzí Laravelu 8.x, bylo nutné si nainstalovat následující prerekvizity:

- PHP verze 7.1.3. a vyšší
- JSON PHP rozšíření
- NODE JS
- Composer – pro řízení knihoven a jejich závislostí

Poté už mi nic nebránilo v instalaci Laravelu pomocí příkazu

```
composer global require „laravel/installer“.
```

Jelikož se jedná o návržení elektronického obchodu, které v budoucnu projde schvalováním a mnohými úpravami, v závislosti na diskusi se zadavateli, vytvořil jsem vše lokálně, což mi umožnila aplikace Xampp Control Panel (viz. Obrázek č. 11). Tato aplikace lokálně spravuje server a databázi zároveň.

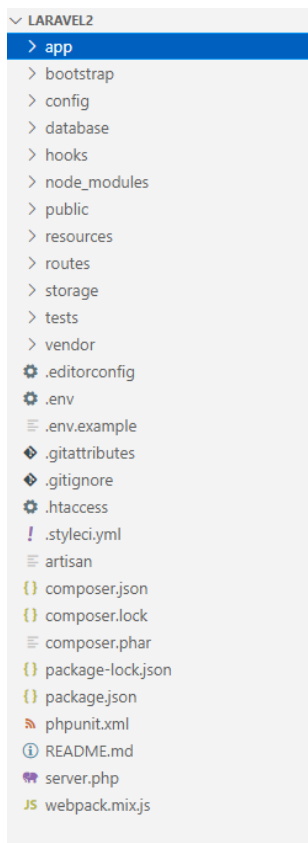


Obrázek č. 9: Xampp Control Panel

(Zdroj: Vlastní zpracování dle aplikace Xampp)

Po konfiguraci jsem si vytvořil první projekt Laravel přes *Composer* příkazem `composer create-project --prefer-dist laravel/laravel Laravel2`.

Příkaz vytvořil následující adresář se spoustou předpřipravených souborů a složek (viz. Obrázek č. 12 vlevo). Na **pravé** části obrázku jsou již finální verze souborů, které jsem vytvořil v průběhu implementace.



**Obrázek č. 10: Adresář projektu Laravel**

(Zdroj: Vlastní zpracování)

V tuto chvíli bylo vše téměř nastaveno, ale protože je lepší mít vždy kód přehlednější a organizovanější, nainstaloval jsem si rozšíření CSS s názvem SASS. Rozšíření není nezbytné k implementaci, ale značně usnadňuje tvorbu, jak je zmíněno v teoretické části. Instalace tohoto rozšíření se provede v příkazové řádce příkazem `npm install`. Tento příkaz vytvoří soubor s názvem `webpack.mix.js`, kde je popsáno propojení SASS s CSS souborem.

### 3.2.2. HTML a CSS

Framework Laravel má pro HTML předpřipravený „šablonovací systém“ zvaný *Blade*, který umožňuje uživateli využívat HTML i PHP ve svých souborech.

V této části jsem se snažil prakticky přetvořit svůj původní návrh do HTML a CSS kódu. V průběhu implementace jsem však došel k mnoha kompromisům a spoustě vylepšení, které se esteticky zdály jako lepší varianta.

Bylo nutné vytvořit kategorii pro každý produkt a oddělit tlačítka v závislosti na informační hodnotě. Nezávisle na požadavcích jsem vytvořil i nové logo, které slouží pouze k představě o velikosti a jeho pozici na stránkách.

Hlavička a zápatí každého HTML dokumentu jsou rozděleny do:

- **<header>** - v tomto bloku se nachází, logo, vyhledávání a tlačítka
- **<main-nav>** - tento blok obsahuje rozdělení do kategorií
- **<footer>** - zápatí stránek

Všechny tyto části jsou součástí bloku `<body>` a třídy `container`, která má nastavený parametr šířky do `960px`, aby zarovнала veškerý obsah na střed strany nehlédě na velikost obrazovky uživatele. Všechny tyto bloky jsou součástí každého HTML dokumentu vytvořeného pro elektronický obchod.

#### **<header>**

Hlavička obsahuje tři části (logo, hledat a tlačítka), které jsem rozdělil pomocí funkcí v SASS `display: flex;` a `justify-content: space-evenly;`, které mi všechny části rozdělily do tří bloků na jednom řádku se stejným odstupem mezi nimi.

Pro pozadí hlavičky jsem vybral přechod odstínu z modré barvy do barvy černé, funkcí `background: linear-gradient(to bottom, #000000 5%, #142a36 100%);`. Tuto funkci jsem využil i u tlačítek, kde jsem také zvolil efekt aktivního překrytí při přejetí kurzorem, čímž zdokonaluji pocit interakce.

## <main-nav>

Stejně jako v hlavičce, i zde jsem rozdělil bloky, pomocí funkce `display: flex;` a `justify-content: space-evenly;`, dvou částí tlačítek kategorií. Kategorie obsahují následující tlačítka:

- První blok – Domů, iPhone, Macbook, Watch, iPad
- Druhý blok – Servis, O nás

V tomto případě jsem tlačítkům přiřadil světlejší přechod barev pro lepší rozlišení.

## <footer>

Do zápatí jsem vypsál veškeré doplňující informace, které se spojují s odkazy na specifické stránky (O nás, Služby, Obchod, Provozní doba, Sledujte nás). Vytvořil jsem i prostor pro sociální sítě, kdyby se v budoucnu zakládaly (viz. Obrázek č. 13). Zápatí také slouží jako zakončení stránky, tudíž jsem ho odlišil stejným přechodem barev jako v hlavičce.

```
<footer id="footer">
  <div class="footer-content container">
    <div class="about-us">O Společnosti
      <ul>
        <li><a href="#" style="text-decoration:none;">Kontakty</a></li>
        <li><a href="#" style="text-decoration:none;">Prodejna</a></li>
        <li><a href="#" style="text-decoration:none;">O nás</a></li>
      </ul>
    </div>
    <div class="services">Služby
      <ul>
        <li><a href="#" style="text-decoration:none;">Autorizovaný servis</a></li>
        <li><a href="#" style="text-decoration:none;">Doprava</a></li>
        <li><a href="#" style="text-decoration:none;">Možnosti platby</a></li>
        <li><a href="#" style="text-decoration:none;">Záruka</a></li>
      </ul>
    </div>
    <div class="services">Obchod
      <ul>
        <li><a href="#" style="text-decoration:none;">Autorizovaný servis</a></li>
        <li><a href="#" style="text-decoration:none;">Doprava</a></li>
        <li><a href="#" style="text-decoration:none;">Možnosti platby</a></li>
        <li><a href="#" style="text-decoration:none;">Záruka</a></li>
      </ul>
    </div>
    <div class="services">Provozní doba
      <ul>
        <li><a href="#" style="text-decoration:none;">Po-Pá: 8:00-18:00</a></li>
        <li><a href="#" style="text-decoration:none;">So-Ne: Zavřeno</a></li>
      </ul>
    </div>
    <div class="follow-us">Sledujte nás
      <ul>
        <li><a href="#" style="text-decoration:none;"><i class="fa fa-globe"></i></a></li>
        <li><a href="#" style="text-decoration:none;"><i class="fa fa-youtube"></i></a></li>
        <li><a href="#" style="text-decoration:none;"><i class="fa fa-github"></i></a></li>
        <li><a href="#" style="text-decoration:none;"><i class="fa fa-twitter"></i></a></li>
      </ul>
    </div>
  </div> <!-- !footer-content -->
</footer>
```

Obrázek č. 11: zdrojový kód zápatí v HTML

(Zdroj: Vlastní zpracování)

Elektronický obchod obsahuje velké množství stránek navržených v HTML, proto popíšu pouze ty nejzajímavější a jejich sekce, které se odlišují.

### 3.2.2.1. Hlavní strana

#### <device-section>

Produkty, jež jsou součástí téměř každé stránky, jsou v části kódu HTML opakovány v závislosti na počtu produktů. Veškeré specifikace těchto produktů jsem, včetně cesty k obrázkům, manuálně napsal, protože se jedná pouze o vizuální a návrhovou část, ve které není napojena databáze, ze které by se data vybrala (viz. Obrázek č. 14).

```
<div class="products text-center">
  <div class="product">
    <div class="picture">
      <a href="#"></a>
    </div>
    <div class="Information">
      <a href="#" style="text-decoration:none;"><div class="product-name">iPhone 6S</div></a>
      <div class="product-price">4990Kč</div>
    </div>
  </div>
  <div class="product">
    <div class="picture">
      <a href="#"></a>
    </div>
    <div class="Information">
      <a href="#" style="text-decoration:none;"><div class="product-name">iPhone 6S</div></a>
      <div class="product-price">4990Kč</div>
    </div>
  </div>
</div>
```

Obrázek č. 12: HTML obsah položky Produkt

(Zdroj: Vlastní zpracování)

### 3.2.2.2. Košík

#### <cart-section>

Košík jsem rozdělil na řádky, do kterých jsem vložil produkt a jeho specifikace. Na pravé straně řádku jsem vytvořil tlačítko pro odstranění položky, výběr počtu položek a na závěr její cenu. Pod těmito řádky jsem dále vytvořil celkovou cenu bez DPH, DPH a celkovou cenu s DPH. Nyní už zbylo pouze vytvořit tlačítka pro pokračování a dokončení nákupu. Vše jsem následně upravil v kódu SASS souboru.

Pod touto sekcí jsem vytvořil náhodné produkty s popisem a cenou, které by se zákazníkům mohly líbit.

### 3.2.2.3. Dokončení objednávky

#### <checkout-section>

Tato sekce se skládá ze dvou hlavních částí:

- **Osobní údaje a platba** – Pro osobní údaje jsem vytvořil vstupy k zadání informací uživatelem (Emailová adresa, Jméno, Příjmení, Ulice, Město, Kraj, Směrovací číslo, Telefonní číslo). K platbě jsem využil již předpřipravený HTML kód obsahující formulář z oficiálních stránek, které se zabývají problematikou online plateb. Toto řešení bude podrobně probráno v dalších kapitolách.
- **Souhrn objednávky** – Obsah této sekce jsem vyřešil velice podobně jako výše zmíněný košík, pouze jsem odebral manuálně upravitelné množství zboží a možnost odebrat zboží.

Následně jsem sekce rozdělil do sloupců, upravil velikosti písma a zarovnal ohraničení vstupů do bloku.

### 3.2.2.4. Servis

#### <price-table>

Celkově se zde jedná o tři tabulky. V první tabulce vypisuji ceny oprav za jednotlivé díly specifických zařízení. Ve druhé tabulce jsem zahrnul popis cen služeb, které se zahrnují do opravy. Ve třetí a zároveň poslední tabulce jsem vypsál jednotlivé kroky, jak postupovat při servisu.

#### <servis>

Zde jsem vytvořil dvě tlačítka, jedno slouží pro výběr položky servisu a druhé pro potvrzení a zobrazení informací o daném servisu. Abych mohl zobrazit informace o servisu zvlášť ve vyskakovacím okně, musel jsem načíst do hlavičky souboru modul JavaScript *Bootstrap*. Propojením parametrů mezi tlačítka a oknem *modal*, jsem byl schopen otevřít vyskakovací okno a dále jej upravovat. Toto okno jsem doplnil o informace o uživateli, názvu servisované položky, popisu závady, stavu objednávky a výpisu cen. Hlavní logiku tomuto oknu dávají dvě tlačítka, kde se uživatel rozhoduje,

zda souhlasí nebo nesouhlasí se servisem zařízení. Na samotném konci tohoto okna je opět stejná implementace platby, která byla zmíněna výše v podkapitole „Dokončení objednávky“.

```
<button type="button5" class="button5 btn-primary" data-toggle="modal" data-target="#exampleModal2">
  Vybrat
</button>
</div>
</div>

<div class="modal fade" id="exampleModal2" tabindex="-1" role="dialog" aria-labelledby="exampleModallabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModallabel">Nová položka servisu</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
```

**Obrázek č. 13: Zdrojový kód HTML: Propojení tlačítka s vyskakovacím oknem**

(Zdroj: Vlastní zpracování)

### 3.2.3. PHP, JavaScript

Ve chvíli, kdy byla část HTML a CSS hotová, vrhl jsem se na tvorbu dynamiky elektronického obchodu.

#### 3.2.3.1. Databáze

Databáze je první krok, jak udělat elektronický obchod dynamický. Níže popíšu, jak jsem postupoval při tvorbě databáze v Laravelu u nejdůležitějších položek. Abych zobrazil data, která jsem do databáze ukládal, využíval jsem phpMyAdmin pro administraci a kontrolu dat.

#### Produkty

Abych mohl data do databáze zadat, musel jsem si vytvořit model dat, který se v Laravelu vytvoří příkazem `php artisan make:model Product -m`. Příkaz mi vytvořil ve složce migrations soubor `date_create_product_table.php`, kde jsem si vypsal veškeré potřebné atributy a datové typy produktu (viz. Obrázek č. 16).

```
class CreateProductsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('products', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('slug');
            $table->string('sorting');
            $table->string('details')->nullable();
            $table->integer('price');
            $table->text('description');
            $table->timestamps();
        });
    }
}
```

Obrázek č. 14: Zdrojový kód atributů tabulky produktu

(Zdroj: Vlastní zpracování)

Následně jsem tyto atributy nahrál do databáze příkazem `php artisan migrate`. Pomocí phpMyAdmin jsem byl schopný si ověřit, jestli se atributy opravdu nahrály do tabulky databáze.



Pomocí Laravelu lze také tabulku naplnit daty, aniž bych je musel manuálně přidávat v phpMyAdmin. Vytvořil jsem si tedy soubor `ProductsTableSeeder.php` příkazem `php artisan make:seed ProductsTableSeeder`. Do tohoto souboru jsem následně vložil veškerá data každého produktu, který bude figurovat v elektronickém obchodě (viz. Obrázek č. 17).

```
class ProductsTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        //iPad Gen8
        //128GB VERZE
        ModelsProduct::create([
            'name' => 'Apple iPad Wi-Fi Gen8 Zlatý',
            'slug' => 'ipad-2020-gold_gen8',
            'sorting' => 'iPad 8.Gen', //
            'details' => '10,2 palců, 128GB',
            'price' => 12999,
            'description' => 'Nádherný 10,2palcový Retina displej.
                            Čip A12 Bionic s Neural Engine.
                            Podporuje Apple Pencil (1. generace) a Smart Keyboard.
                            8MP zadní fotoaparát, 1,2MP přední fotoaparát FaceTime HD
                            Stereo reproduktory.
                            Wi-Fi 802.11ac a 4G LTE Advanced na mobilní data.
                            Výdrž baterie až 10 hodin.
                            Konektor Lightning na nabíjení a připojení doplňků.
                            Touch ID na bezpečné ověřování a Apple Pay.
                            iPadOS 14 přináší nové možnosti navržené přímo pro iPad.',
```

**Obrázek č. 15: Zdrojový kód dat produktu**

(Zdroj: Vlastní zpracování)

V posledním kroku opět stačilo nahrát tato data do databáze příkazem `php artisan make:seed`. Tímto byla data v databázi připravena k dalšímu užívání.

## Kategorie

Každý typ zařízení má své kategorie podle modelu, proto jsem si vytvořil další tabulku s názvem „Category“, která obsahuje název a údaj k řazení.

Vytvoření souborů a uložení dat do databáze mělo stejný průběh jako u výše zmíněných produktů. Změna proběhla pouze u názvu a obsahu tabulky (viz. Obrázek č. 18).

```
class CreateCategoriesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('category', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('sorting');
            $table->timestamps();
        });
    }
}
```

Obrázek č. 16: Zdrojový kód atributů tabulky kategorií

(Zdroj: Vlastní zpracování)

## Kategorie a Produkty

Aby filtrování produktů podle kategorií mohlo fungovat, vytvořil jsem třetí tabulku, která mi jasně určí spojitost mezi kategoriemi a produkty.

Opět jsem se držel stejného postupu vytvoření modelu a migrace, jako v předešlých tabulkách, pouze v souboru migrace jsem vytvořil spojitost mezi tabulkami za pomoci vedlejších klíčů (viz. Obrázek č. 19). Poté jsem přidal každému produktu specifickou kategorii a tím jsem dokončil propojení tabulek.

```
public function up()
{
    Schema::create('category_product', function (Blueprint $table) {
        $table->id();
        $table->integer('product_id')->unsigned()->nullable();
        $table->foreign('product_id')->references('id')->on('products')->onDelete('cascade');
        $table->integer('category_id')->unsigned()->nullable();
        $table->foreign('category_id')->references('id')->on('category')->onDelete('cascade');
        $table->timestamps();
    });
}
```

Obrázek č. 17: Zdrojový kód propojení tabulek

(Zdroj: Vlastní zpracování)

## Servis

Stejně jako u tabulek produktů a kategorií jsem vytvořil tabulku pro servis, kde se budou zaznamenávat veškeré opravy s vazbou na tabulku uživatele. Do tabulky jsem zatím nenahrával žádná data o servisech, pouze jsem vytvořil migraci s atributy (viz. Obrázek č. 20). Tato problematika bude řešena dále v práci.

```
public function up()
{
    Schema::create('opravies', function (Blueprint $table) {
        $table->increments('id');
        $table->string('email');
        $table->string('device');
        $table->string('defect');
        $table->string('state');
        $table->string('priceYes');
        $table->string('priceNo');
        $table->timestamps();
    });
}
```

Obrázek č. 18: Zdrojový kód atributů tabulky oprav

(Zdroj: Vlastní zpracování)

## Uživatelé

Tabulka uživatelů s možností obnovení hesla a autentizací jsou již součástí Laravelu, proto se automaticky uloží do databáze současně s ostatními tabulkami. Do této tabulky budu ukládat informace o registrovaných uživateli.

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

Obrázek č. 19: Zdrojový kód atributů tabulky uživatelů

(Zdroj: Vlastní zpracování)

### 3.2.3.2. PHP

Předtím než jsem postupně začal nahrávat data z databáze, bylo vhodné, za pomoci funkcí Laravelu, kód v souborech zpřehlednit, aby nevznikaly duplicity. Hlavička a zápatí se objevují v každém PHP souboru, proto jsem si vytvořil soubor s názvem `layout.blade.php`, kde jsem tyto části vložil a z ostatních dokumentů smazal. Mezi bloky hlavičky a zápatí stačilo vložit funkci `@yield()`, která importuje veškerý obsah, označený z jiných dokumentů. V ostatních dokumentech jsem tedy rozšířil záhlaví `@extends()` a obsah ohraničil funkcí `@section()`, `@endsection`.

Za normálních okolností nelze načítat data z databáze přímo v souborech `blade`. K tomu slouží, jako prostředník, řadič (*Controller*), který řeší aplikační logiku. Tento řadič jsem tedy vytvořil pro všechny soubory `blade`, které využívají tuto logiku, příkazem `php artisan make:controller Name`. Každá funkce v těchto řadičích má návratovou hodnotu, kterou získávám pomocí cest v souboru `web.php` a http požadavků (`get`, `post`, `patch`, `delete`).

#### Produkt

Produkt se nachází ve více souborech. V každém má odlišné parametry a načítání z databáze probíhá jinak, proto se zdrojový kód v řadičích liší.

Řadič pro úvodní stránku je velice zjednodušený, protože stránka obsahuje všechny produkty, které jsou náhodně zobrazeny (viz. Obrázek č. 22).

```
public function show($slug)
{
    //slug equal to slug
    $product= Product::where('slug', $slug)->firstOrFail();
    $mightAlsoLike= Product::where('slug', '!=', $slug)->mightAlsoLike()->take(4)->get();

    return view('product')->with([
        'product'=> $product,
        'mightAlsoLike' => $mightAlsoLike,
    ]);
}
```

Obrázek č. 20: Řadič úvodní stránky

(Zdroj: Vlastní zpracování)

Řadiče stránek produktů Macbook, iPad, iPhone, Watch, jsou komplexnější, protože jsou součástí jedné tabulky a zároveň se třídí do kategorií. Přesto se tyto stránky od sebe příliš neliší, proto jsem v řadičích změnil pouze náležité hodnoty z databáze.

```
public function index()
{
    if(request()->category){
        $products = Product::with('categories')->whereHas('categories', function($query){
            $query->where('sorting', request()->category);
        })->get();
        $categories = Category::where('id', '10')
            ->orWhere('id', '11')
            ->orWhere('id', '12')
            ->orWhere('id', '13')
            ->get();
    } else{
        $products = Product::where('sorting', 'Apple Watch Nike')
            ->orWhere('sorting', 'Apple Watch SE')
            ->orWhere('sorting', 'Apple Watch 6')
            ->orWhere('sorting', 'Apple Watch 3')
            ->inRandomOrder()
            ->get();
        $categories = Category::where('id', '10')
            ->orWhere('id', '11')
            ->orWhere('id', '12')
            ->orWhere('id', '13')
            ->get();
    }

    $data = [
        'products' => $products,
        'categories' => $categories
    ];

    return view('watch')->with($data);
}
```

**Obrázek č. 21: Zdrojový kód řadiče pro stránky s Apple Watch**

(Zdroj: Vlastní zpracování)

Poté co bylo vše připraveno, postupně jsem vložil data do produktů v souborech blade. Duplicity jsem zredukoval cyklem @foreach a doplnil odkazy na přesměrování ze souboru web.php.

```
<div class="products text-center">
    @foreach ($products as $product)
        <div class="product">
            <div class="picture">
                <a href="{{route('welcome.show', $product->slug)}}">
                </a>
            </div>

            <div class="Information">
                <a href="{{route('welcome.show', $product->slug)}}" style="text-decoration:none;">
                <div class="product-name">{{ $product->name }}</div></a>
                <div class="product-price">{{ $product->presentPrice()}}</div>
            </div>
        </div>
    @endforeach
</div>
```

**Obrázek č. 22: Zdrojový kód cyklu produktů a přiřazení dat**

(Zdroj: Vlastní zpracování)

## Košík

Implementace košíku dokáže být složitá a časově náročná záležitost, proto jsem využil volně dostupný balíček, který se nahraje do Laravelu příkazem `composer require hardevine/shoppingcart`. Balíček si vytvoří soubor, ve kterém lze upravovat veškeré jeho funkce dle potřeby uživatele. Funkce tohoto balíčku umí snadno pracovat s polem, do kterého jsem v řadiči ukládal informace o produktu. Dále jsem v řadiči definoval notifikace, pokud bylo zboží přidáno do košíku nebo se už v košíku nachází.

```
public function store(Request $request)
{
    $duplicates = Cart::search(function($cartItem, $rowId) use ($request){
        return $cartItem->id === $request->id;
    });

    if($duplicates->isNotEmpty()){
        return redirect()->route('cart.shop')->with('success-message', 'Toto zboží už je v košíku');
    }

    Cart::add($request->id, $request->name, 1, $request->price)
    ->associate('App\Models\Product');

    return redirect()->route('cart.shop')->with('success-message', 'Zboží bylo přidáno do košíku');
}
```

Obrázek č. 23: Zdrojový kód řadiče stránky Košíku

(Zdroj Vlastní zpracování)

U všech stránek, s možností přidání produktu do košíku, jsem využil pro předávání parametrů produktu, právě jednu z funkcí balíčku, která parametry dále posílá na stránky s košíkem. V souboru košíku `blade.php` jsem umístil řádky produktů do cyklu `@foreach`, aby se automaticky doplnily při přidávání dalších produktů a automaticky se přepočítávalo jejich množství.

Znemožnil jsem mnohonásobné přidávání stejného produktu do košíku a vyřešil problematiku možností výběru množství přímo na stránkách košíku, kde bude cena automaticky přepočítána.

Na závěr jsem přidal funkce pro vypočítání celkové ceny bez DPH, DPH a celkové ceny s DPH.

## Přihlášení a registrace

Obě stránky jsou automaticky vytvořeny, se všemi komponenty databáze, už při instalaci Laravelu. Pouze pro vytvoření autentizace a souborů `blade.php`, jsem musel využít příkazy:

- `composer require laravel/ui`
- `php artisan ui vue --auth`
- `php artisan make:auth`

Poté už stačilo pouze přidat styly používané v jiných dokumentech a registrace s přihlášením byly plně funkční.

## Dokončení objednávky

Na stránce dokončení objednávky jsem využil přehled košíku, který prakticky přebírá informace z výše uvedené implementace košíku.

Součástí požadavků byla i platba kartou. Zde jsem použil dokumentaci oficiálních stránek Stripe. Na těchto stránkách je vytvořena kompletní a volně šiřitelná implementace platby karet, kterou si musí každý uživatel modifikovat v závislosti na svém řešení. Aby platba mohla fungovat, je nutné mít dva takzvané „*API keys*“:

- **Publishable key** - slouží k identifikaci účtu pomocí Stripe, může být zveřejněn
- **Secret key** – může uskutečnit jakýkoliv požadavek Stripe bez omezení, za žádných okolností by neměl být zveřejněn

Stručně řečeno, slouží ke komunikaci karty v elektronickém obchodě a serverem Stripe. Tyto klíče jsou veřejně dostupné na mnoha stránkách, které se uživateli zobrazí po registraci. (16)

Z mého hlediska bylo nejvýhodnější využívat Stripe z důvodu velkého množství funkcí, které nabízí, především kvůli monitorování všech plateb, grafické znázornění příjmů a velkému množství doplňujících informací, které lze předávat ze stránek s platbou. Dalším důvodem je dostupnost služby, protože známější služby vyžadují komunikaci s bankou, kterou zajímá výše obrátu společnosti.

V implementaci souboru `blade.php`, jsem tedy doplnil kód jazyka HTML a JavaScript, do kterého jsem předal vstupní parametry. Tyto parametry uživatel na stránce vyplní a

při potvrzení platby vše odešle do Stripe, čímž dokončí objednávku. K odeslání veškerých informací, jsem nastavil řadič stránky a také parametr ceny, aby odpovídal celkové ceně v košíku (viz. Obrázek č. 26).

```
public function paymentProcess(Request $request)
{
    //header('Content-Type: application/json');
    //require 'vendor/autoload.php';
    $contents = Cart::content()->map(function($item){
        return $item->model->slug.'- '.$item->qty;
    }->values()->toJson());

    \Stripe\Stripe::setApiKey('');

    try {

        // retrieve JSON from POST body

        $json_str = $request->getContent();
        $json_obj = json_decode($json_str);

        $paymentIntent = \Stripe\PaymentIntent::create([
            'amount' => $this->calculateOrderAmount($json_obj->items),
            'currency' => 'czk',
            'metadata' => [
                'contents' => $contents,
                'quantity' => Cart::instance('default')->count(),
            ],
        ]);

        $output = [
            'clientSecret' => $paymentIntent->client_secret,
        ];

        Cart::instance('default')->destroy();
        return response()->json($output);

    } catch (\Error $e) {
        http_response_code(500);

        return response()->json(['error' => $e->getMessage()]);
    }
}

public function calculateOrderAmount(array $items): int {
    $myprice = Cart::total()*100;

    return $myprice;
}
```

**Obrázek č. 24: Zdrojový kód předání parametrů platební karty**

(Zdroj: Vlastní zpracování)

Dokončení objednávky jsem zpřístupnil pouze pro uživatele, kteří jsou zaregistrovaní a jejich přihlašovací email je automaticky doplněn v kolonce „Emailová Adresa“.

Pro testování plateb jsem využil předdefinované karty, jelikož se jedná pouze o testovací rozhraní, nikoliv o online rozhraní.



## Servis

Servis jsem navrhl s logikou přístupu anonymního uživatele, který si může procházet ceník oprav a návod, jak postupovat při opravě. Pokud uživatel má zařízení v opravě, musí se přihlásit pro zobrazení jejího detailu a stavu. Této funkčnosti jsem dosáhl podmínkou `@if(auth()->user())`. Dále jsem přiřadil přihlášenému uživateli servis, který se aktuálně vyskytuje v databázi. Pokud neexistuje žádný servis, který by byl přiřazený k uživateli, neobjeví se možnost výběru položky. Informace o těchto opravách zadává vždy administrátor manuálně do systému.

Při výběru položky a stisknutí tlačítka se objeví vyskakovací okno, které obsahuje výpis informací z databáze oprav a dvě tlačítka. Platba kartou byla implementována podobným způsobem jako ve výše uvedeném „Dokončení objednávky“. Rozdílem byla implementace v jazyce JavaScript, kde jsem vložil podmínku pro zobrazení platby kartou, po stisknutí jednoho ze dvou tlačítek. V závislosti na stisknutém tlačítku se provede i přepočet vybrané ceny.

```
<script>
    jQuery(function() {

        $('#YesButton').click(function(){
            DoStripe(document.getElementById('Ano').value);
            $('#Stripik').removeClass('d-none');
        });

        $('#NoButton').click(function(){
            DoStripe(document.getElementById('Ne').value);
            $('#Stripik').removeClass('d-none');
        })

        function DoStripe(amount){
```

Obrázek č. 25: Zdrojový kód podmínka výběru ceny v jazyce JavaScript

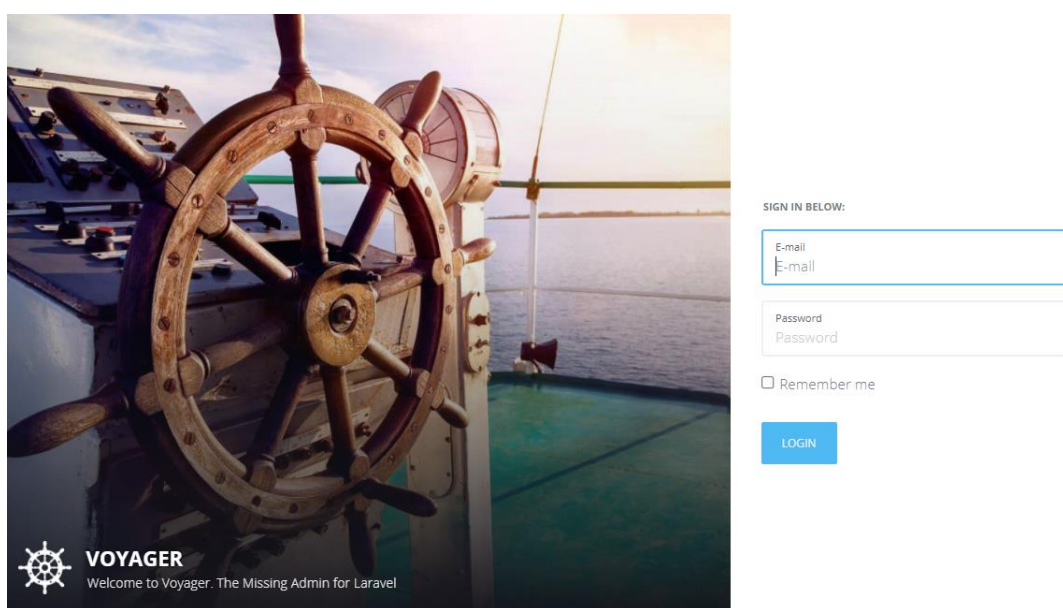
(Zdroj: Vlastní zpracování)

### 3.3. Administrátor

#### Voyager

Pro rozhraní administrátora, jsem využil nástroj Voyager. V dokumentaci jej popisují následujícím způsobem: „Voyager je jednoduše rozšíření admin pro vaši Laravel aplikaci“. (17)

Tento způsob jsem zvolil z důvodu jednoduchosti a organizace celého prostředí. Administrátor se přihlásí svými údaji do nástroje Voyager a dostane se na přehled celé databáze pro elektronický obchod.



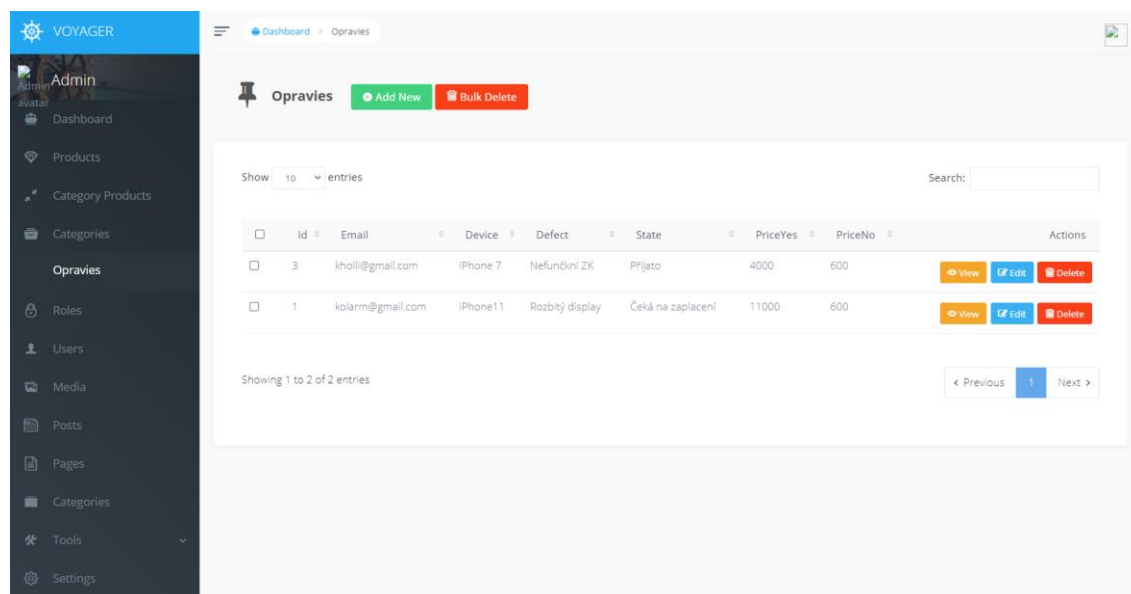
**Obrázek č. 26: Přihlašovací stránky nástroje Voyager**

(Zdroj: Vlastní zpracování dle <https://voyager.devdojo.com/>)

Dále může v tabulkách přidávat nebo odebírat položky, tabulky a upravovat hodnoty položek (viz. Obrázek č. 29). Voyager má na rozdíl od administrace v phpMyAdmin mnoho výhod například:

- Libovolné řazení tabulek
- Možnost označení tabulek ikonami
- Editace práv a vstupů atributů
- Nastavení validace atributů

Odezva stránek je prakticky okamžitá, obsah je intuitivní a úpravy položek se ihned projeví v elektronickém obchodě.

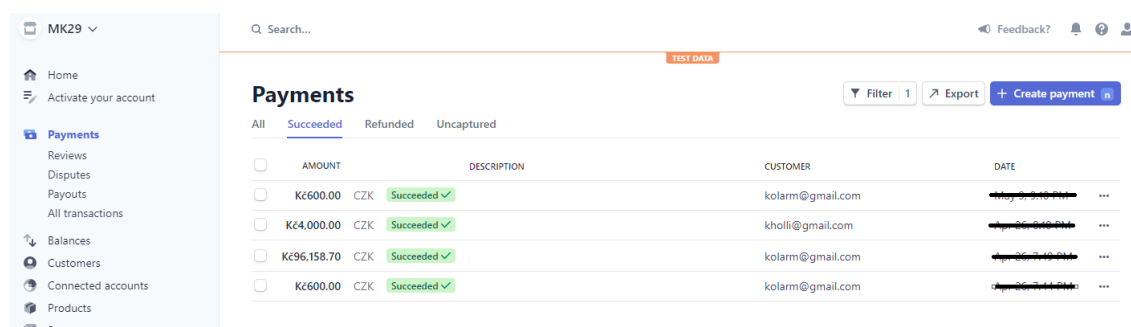


Obrázek č. 27: Přehled tabulek databáze s detailem na tabulku oprav

(Zdroj: Vlastní zpracování)

## Stripe

Stripe jsem již zmiňoval v části implementace plateb. Nástroj ovšem neslouží pouze pro přehled plateb. Elektronický obchod jsem navrhl tak, že kromě platebních údajů se předávají informace o uživateli, druhu servisu nebo položkách nákupu a jejich množství. Díky těmto informacím lze rozlišit, o jakou službu se jedná a následně odeslat fakturu a expedovat zboží. V případě servisu se odešle faktura a uzavře položka opravy. Nástroj nám také ukazuje přehledy všech plateb a grafy tržeb za určitá období.



Obrázek č. 28: Přehled plateb v nástroji Stripe

(Zdroj: Vlastní zpracování dle stránek <https://dashboard.stripe.com/>)

### **3.4. Uživatelské rozhraní**

Tato se část bude věnovat popisu stránek elektronického obchodu a funkčnosti z hlediska uživatele.

#### **3.4.1. Úvodní strana**

Na úvodní straně se uživateli zobrazí menu s tlačítky, obsah s nabízenými produkty a zápatí s dodatečnými informacemi (viz. Příloha 1). Tato stránka je dosažitelná taktéž z tlačítka „Domů“ a loga v menu.

Každý produkt obsahuje odkaz, který uživatele po kliknutí přesměruje na jeho detailní informace a nabídne možnost přidání produktu do košíku (viz. Příloha 2).

#### **3.4.2. Registrace a přihlášení**

Pokud uživatel nemá vytvořený účet, musí se nejprve registrovat, aby mu byla zpřístupněna možnost dokončení nákupu, anebo zobrazení servisované položky. Zadávané údaje jsou kontrolovány a hlásí chybu při nesprávném vyplnění.

Po přihlášení je uživatel přesměrován na původní stránku, na které se nacházel před přihlášením.

#### **3.4.3. iPhone, Macbook, Watch, iPad**

Každá z těchto stránek obsahuje svou kategorii produktů a nabízí možnost jejich filtrování podle modelu. Stejně jako na úvodní straně i zde je možné zobrazení detailů produktu a přidání do košíku (viz. Příloha 3).

#### **3.4.4. Košík**

Košík obsahuje zboží, které bylo vybráno uživatelem. Pokud je košík prázdný, nabídne pouze tlačítko „Pokračovat v nákupu“ a produkty, které by mohly zaujmout. U každé položky je možná změna množství nebo její odebrání. Následně je přepočítána cena s DPH a zpřístupní se tlačítko „Dokončit objednávku“ (viz. Příloha 4).

### **3.4.5. Dokončení objednávky**

U dokončení objednávky se v prvním sloupci zobrazí políčka pro vyplnění údajů uživatele a platební údaje. Ve druhém sloupci zase figuruje přehled objednávky z košíku s celkovou částkou k úhradě (viz. Příloha 5).

Po vyplnění údajů a potvrzení tlačítka „Zaplatit a dokončit objednávku“ proběhne dokončení a přesměrování na stránku s informací o dokončení objednávky.

### **3.4.6. Servis**

Pokud si stránky se servisem prohlíží anonymní uživatel, zobrazí se mu pouze tabulky s ceníky služeb a návodem, jak postupovat při servisu.

Pokud má přihlášený uživatel zrovna servis v procesu, zpřístupní se mu možnost výběru této položky s tlačítkem „Vybrat“. Po stisknutí tlačítka se objeví vyskakovací okno s informacemi týkající se servisu, včetně závady a stavu servisu. Uživatel tedy může pravidelně kontrolovat tento stav a je notifikován, když proběhne změna. Pokud je servis ve stavu „Čeká na zaplacení“, uživateli se zobrazí návrh cen, jestli souhlasí s opravou nebo ne. Pokud uživatel nesouhlasí s plnou cenou za servis, musí zaplatit alespoň částku za diagnostiku. Po vybrání jakékoliv z možností se zobrazí platební údaje a po jejich vyplnění a potvrzení tlačítka „Zaplatit a dokončit servis“, je uživatel přesměrován na stránku s informací o dokončení servisu. (viz. Příloha 6)

### **3.4.7. O nás**

Tato stránka obsahuje adresu, kontaktní údaje na zaměstnance společnosti a otevírací dobu. Pro lepší orientaci se na stránce nachází i mapa s lokalitou pobočky.

### **3.5. Ekonomické zhodnocení**

O vzniku každého elektronického obchodu na míru, rozhoduje nejdůležitější faktor celkové ceny. Pro realizaci elektronického obchodu v této práci byly využity volně dostupné programy a nástroje. Z tohoto důvodu je jediným nákladem čas strávený na tvorbě tohoto elektronického obchodu. Tato částka je rozdělena v závislosti na komplexitě řešení. V této práci elektronický obchod obsahuje základní funkcionalitu s nadstandardní službou, proto se řadí do kategorie středního webu a cena se odvíjí od splněných kritérií. Celková realizace projektu zabrala přibližně 70 hodin, včetně konzultace s majiteli společnosti. Za předpokladu hodinového ohodnocení 300 Kč, by celkové náklady byly 21 000 Kč. Pokud by se společnost rozhodla přejít na jiná řešení platby, přechod na placený hosting a administrační nástroj, náklady by vzrostly.

Optimalizaci a správu elektronického obchodu zvládne jakýkoliv zaměstnanec společnosti XYZ, díky přívětivému rozhraní ve Voyager, proto není potřeba najímat nového zaměstnance na pozici programátora.

## 4. Závěr

Hlavním cílem této bakalářské práce bylo navržení elektronického obchodu společnosti XYZ dle specifických požadavků a vytvoření moderního řešení určitých služeb, které pomohou ke zvýšení počtu zákazníků a jejich spokojenosti.

V úvodu mé práce je popsán teoretický základ jazyků a metod, potřebných pro vytvoření elektronického obchodu od samotného začátku, který je nezbytný k objasnění pojmů v dalších částech práce. Dále jsem se věnoval požadavkům na elektronický obchod a analýze společnosti XYZ. V analýze bylo důležité objasnit, jaké má společnost silné a slabé stránky a jaké konkurenci čelí na trhu. V závislosti na zjištěných informacích jsem pracoval na diverzifikaci alespoň určité části elektronického obchodu.

Praktická část byla zaměřena na kompletní tvorbu elektronického obchodu. Vše bylo vytvořeno pro co nejlepší estetický dojem, jednoduchost a intuitivnost. Především jsem se snažil odlišit od jiných elektronických obchodů při zpracování servisové části, administrátorského rozhraní a nástroje pro správu plateb, na což společnost kladla velký důraz.

Díky tomu, že byl realizován elektronický obchod v závislosti na analytické části se specifikacemi, které společnost stanovila, lze konstatovat, že cíle práce byly splněny.

## Seznam použité literatury

- (1) GRÖPL, Tomáš. *HTML, CSS a JavaScript: Referenční příručka*. Praha: BEN - technická literatura, 2002. ISBN 9788073000998.
- (2) PÍSEK, Slavoj. *HTML: začínáme programovat*. 3., aktualiz. vyd. [i.e.] 1. vyd. Praha: Grada, 2010. Průvodce (Grada). ISBN 978-80-247-3117-9.
- (3) HOGAN, Brian P. *HTML5 and CSS3: develop with tomorrow's standards today*. Dallas: Pragmatic Bookshelf, c2010. Pragmatic programmers. ISBN 978-1-93435-668-5.
- (4) PROKOP, Marek. *CSS pro webdesignery*. Vyd. 2. Brno: CP Books, 2005. ISBN 80-251-0487-7.
- (5) SNÍŽEK, Pavel. *CSS pro zelenáče*. Praha: Neocortex, 2004, c2003. Pro zelenáče. ISBN isbn80-86330-14-1.
- (6) SASS: *Documentation* [online]. [cit. 2021-5-2]. Dostupné z: <https://sass-lang.com/documentation>
- (7) VÁCLAVEK, Petr. *JavaScript: hotová řešení*. Brno: Computer Press, 2003. K okamžitému použití (Computer Press). ISBN 80-7226-854-6.
- (8) WELLING, Luke a Laura THOMSON. *Mistrovství PHP a MySQL*. Přeložil Ondřej BAŠE. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1.
- (9) CAMERON, Dane. *HTML5, JavaScript, and jQuery: 24-hour trainer*. Indianapolis: Wrox, [2015]. ISBN 9781119001164.
- (10) OTWELL, Taylor. *Laravel: Documentation* [online]. [cit. 2021-5-5]. Dostupné z: <https://laravel.com/docs/8.x>
- (11) phpMyAdmin devel team Revision f713094c. *PhpMyAdmin: Documentation* [online]. [cit. 2021-5-5]. Dostupné z: <https://docs.phpmyadmin.net/en/latest/>
- (12) CASTAGNETTO, Jesus. *PHP programujeme profesionálně*. 2., opr. a aktualiz. vyd. Přeložil Luděk ROUBÍČEK. Praha: Computer Press, 2004. Internet. ISBN 80-7226-310-2.



- (13) STEINOVÁ, Martina, Michal PŘÁDKA a Miluše HLUCHNÍKOVÁ. *E-marketing II.: marketingová komunikace na internetu: elektronické obchodování*. Ostrava: Vysoká škola báňská - Technická univerzita, 2003. ISBN isbn80-248-0351-8.
- (14) GRASSEOVÁ, Monika, Radek DUBEC a David ŘEHÁK. *Analýza v rukou manažera: 33 nejpoužívanějších metod strategického řízení*. Brno: Computer Press, 2010. ISBN isbn978-80-251-2621-9.
- (15) CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN isbn978-80-251-2328-7.
- (16) *Stripe: Documentation* [online]. [cit. 2021-5-9]. Dostupné z: <https://stripe.com/docs/payments/integration-builder>
- (17) *Voyager: Documentation* [online]. [cit. 2021-5-9]. Dostupné z: <https://voyager-docs.devdojo.com/>

## Seznam obrázků

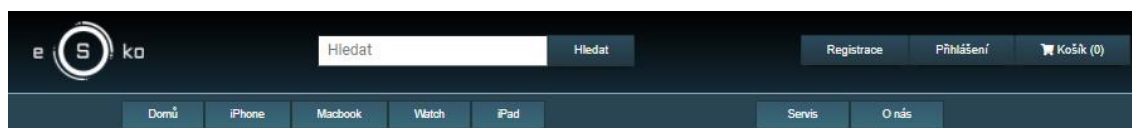
<b>Obrázek č. 1: Struktura HTML</b> (Zdroj: Vlastní zpracování) .....	14
<b>Obrázek č. 2: Rozdíly CSS a HTML</b> (Zdroj: 4, s. 14, 15).....	15
<b>Obrázek č. 3: Rozdíly SASS (vlevo) a CSS (vpravo)</b> (Zdroj: Vlastní zpracování) ....	18
<b>Obrázek č. 4: Pohled</b> (Zdroj: 10) .....	21
<b>Obrázek č. 5: phpMyAdmin rozhraní</b> (Zdroj: 11).....	22
<b>Obrázek č. 6: Počet elektronických obchodů, zaměřených na prodej elektroniky v ČR</b> (Zdroj: <a href="https://www.ceska-ecommerce.cz/#reseni">https://www.ceska-ecommerce.cz/#reseni</a> ) .....	25
<b>Obrázek č. 7: Návrh ERD elektronického obchodu</b> (Zdroj: Vlastní zpracování) .....	31
<b>Obrázek č. 8: Návrh designu hlavní strany</b> (Zdroj: Vlastní zpracování) .....	32
<b>Obrázek č. 9: Xampp Control Panel</b> (Zdroj: Vlastní zpracování dle aplikace Xampp) .....	33
<b>Obrázek č. 10: Adresář projektu Laravel</b> (Zdroj: Vlastní zpracování) .....	34
<b>Obrázek č. 11: zdrojový kód zápatí v HTML</b> (Zdroj: Vlastní zpracování) .....	36
<b>Obrázek č. 12: HTML obsah položky Produkt</b> (Zdroj: Vlastní zpracování).....	37
<b>Obrázek č. 13: Zdrojový kód HTML: Propojení tlačítka s vyskakovacím oknem</b> (Zdroj: Vlastní zpracování).....	39
<b>Obrázek č. 14: Zdrojový kód atributů tabulky produktu</b> (Zdroj: Vlastní zpracování) .....	40
<b>Obrázek č. 15: Zdrojový kód dat produktu</b> (Zdroj: Vlastní zpracování) .....	41
<b>Obrázek č. 16: Zdrojový kód atributů tabulky kategorií</b> (Zdroj: Vlastní zpracování) .....	42
<b>Obrázek č. 17: Zdrojový kód propojení tabulek</b> (Zdroj: Vlastní zpracování).....	42
<b>Obrázek č. 18: Zdrojový kód atributů tabulky oprav</b> (Zdroj: Vlastní zpracování) ..	43
<b>Obrázek č. 19: Zdrojový kód atributů tabulky uživatelů</b> (Zdroj: Vlastní zpracování) .....	43
<b>Obrázek č. 20: Řadič úvodní stránky</b> (Zdroj: Vlastní zpracování).....	44

<b>Obrázek č. 21: Zdrojový kód řadiče pro stránky s Apple Watch</b> (Zdroj: Vlastní zpracování).....	45
<b>Obrázek č. 22: Zdrojový kód cyklu produktů a přiřazení dat</b> (Zdroj: Vlastní zpracování).....	45
<b>Obrázek č. 23: Zdrojový kód řadiče stránky Košíku</b> (Zdroj Vlastní zpracování) ....	46
<b>Obrázek č. 24: Zdrojový kód předání parametrů platební karty</b> (Zdroj: Vlastní zpracování).....	48
<b>Obrázek č. 25: Zdrojový kód podmínka výběru ceny v jazyce JavaScript</b> (Zdroj: Vlastní zpracování).....	49
<b>Obrázek č. 26: Přihlašovací stránky nástroje Voyager</b> (Zdroj: Vlastní zpracování dle <a href="https://voyager.devdojo.com/">https://voyager.devdojo.com/</a> ) .....	50
<b>Obrázek č. 27: Přehled tabulek databáze s detailem na tabulku oprav</b> (Zdroj: Vlastní zpracování) .....	51
<b>Obrázek č. 28: Přehled plateb v nástroji Stripe</b> (Zdroj: Vlastní zpracování dle stránek <a href="https://dashboard.stripe.com/">https://dashboard.stripe.com/</a> ).....	51

## Seznam příloh

<b>Příloha 1: Úvodní strana elektronického obchodu .....</b>	<b>61</b>
<b>Příloha 2: Detail produktu .....</b>	<b>62</b>
<b>Příloha 3: Filtr modelu iPhone XR .....</b>	<b>63</b>
<b>Příloha 4: Košík .....</b>	<b>64</b>
<b>Příloha 5: Dokončení objednávky .....</b>	<b>65</b>
<b>Příloha 6: Položka servisu .....</b>	<b>66</b>

## Příloha 1: Úvodní strana elektronického obchodu



### Nabízené produkty



Apple Watch S6 Červený  
12,290 Kč



Apple iPhone XR Černý  
15,990 Kč



Apple iPhone 11 Zelený  
17,990 Kč



Apple Watch S6 Černý  
12,290 Kč



Macbook Pro 16 i9 1TB  
77,990 Kč



Apple iPhone 12 PRO Stříbrný  
29,990 Kč



Apple iPhone 11 Žlutý  
17,990 Kč



Apple iPhone XR Červený  
15,990 Kč



Macbook Air 13 M1 256GB  
29,990 Kč



Apple iPad Wi-Fi Gen8 Šedý  
12,990 Kč



Macbook Air 13 i5 256GB  
29,990 Kč



Apple iPad Air Wi-Fi Modrý  
16,990 Kč

#### © Michal Kolář

- [skolard4](#)
- [187436](#)

#### O Společnosti

- [Kontakty](#)
- [Prodejna](#)
- [O nás](#)

#### Služby

- [Autorizovaný servis](#)
- [Doprava](#)
- [Možnosti platby](#)
- [Záruka](#)


#### Provozní doba

- [Po-Pá: 8:00-18:00](#)
- [So-Ne: Zavřeno](#)

#### Sledujte nás

- [Facebook](#)
- [Instagram](#)
- [Twitter](#)

## Příloha 2: Detail produktu



[Hledat](#)

[Registrace](#)

[Přihlášení](#)

[Košík \(0\)](#)

[Domů](#)

[iPhone](#)

[Macbook](#)

[Watch](#)

[iPad](#)

[Servis](#)

[O nás](#)



### Apple Watch S6 Červený

44 mm

12,290 Kč

Budoucnost zdraví na vaší ruce – chytré hodinky Apple Watch Series 6. Pouzdro o velikosti 44 mm. Stále zapnutý Retina displej. GPS. Saturace kyslíkem – hladina kyslíku v krvi. Měření EKG. Oznámení o pomalém a rychlém tepu. Oznámení o nepravdělném srdečním rytmu. Voděodolnost do 50 metrů. Detekce pádu. Kompas a výškoměr. SOS funkce. Monitorování hluku. Čip S6 SiP s dvoujádrovým procesorem. Reprodukční a mikrofon druhé generace.

[Do košíku](#)

Mohlo by se vám líbit...



Apple Watch S6 Bílý  
12,290 Kč



Apple iPad mini 5 Wi-Fi Šedý  
11,490 Kč



Apple Watch SE Černý  
8,790 Kč



Apple iPhone 12 Černý  
24,490 Kč

#### © Michal Kolář

- skolar64
- 187436

#### O Společnosti

- Kontakty
- Prodejna
- O nás

#### Služby

- Autorizovaný servis
- Doprava
- Možnosti platby
- Záruka

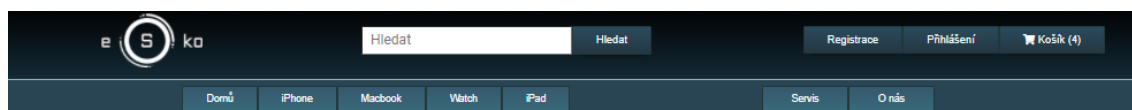
#### Provozní doba

- Po-Pá: 8:00-18:00
- So-Ne: Zavřeno

#### Sledujte nás

- 
- 
- 

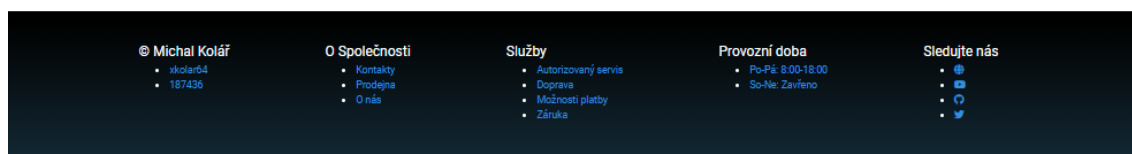
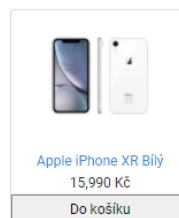
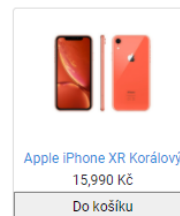
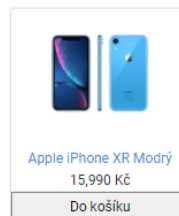
### Příloha 3: Filtr modelu iPhone XR



## iPhone

### Kategorie

- iPhone 11
- iPhone SE
- iPhone XR
- iPhone 12
- iPhone 12 Pro
- iPhone 12 mini
- iPhone 12 Pro Max



## Příloha 4: Košík

  Hledat

Registrace Přihlášení Košík (4)

Domů iPhone Macbook Watch iPad Servis O nás

Zboží bylo přidáno do košíku

### V košíku jsou 4 zařízení



**Apple iPhone 12 Mini Modrý**  
5,4 palců, 128GB

Odebrat  42,980 Kč



**Apple iPhone 11 Zelený**  
6,1 palců, 64GB

Odebrat  17,990 Kč



**Apple Watch SE Černý**  
4,4 mm

Odebrat  8,790 Kč

Doprava zdarma

Celkem: 69,760 Kč  
DPH(21%): 14,650 Kč  
**Celkem(s DPH): 84,410 Kč**

[Pokračovat v nákupu](#)

[Dokončit objednávku](#)

Mohlo by se vám líbit...



[Apple iPhone 12 PRO Max Šedý](#)  
33,990 Kč



[Apple iPad Wi-Fi Gen8 Stříbrný](#)  
12,999 Kč




[Apple iPhone 12 PRO Max Stříbrný](#)  
33,990 Kč



[Apple Watch Nike S6 Černý](#)  
12,290 Kč



## Příloha 5: Dokončení objednávky



### Doprava a platba

#### Osobní údaje

Emailová Adresa

kolarm@gmail.com

Jméno a Příjmení

Ulice

Město

Kraj

Směrovací číslo

Telefonní číslo

#### Platební údaje

Jméno na kartě

 Číslo karty

MM / RR CVC

#### Vaše objednávka



Apple iPhone 12 Mini Modrý  
5,4 palců, 128GB  
21,490 Kč

2



Apple iPhone 11 Zelený  
6,1 palců, 64GB  
17,990 Kč

1



Apple Watch SE Černý  
44 mm  
8,790 Kč

1

Cena bez DPH

69,760 Kč

DPH

14,650 Kč

**Celkem s DPH**

**84,410 Kč**

© Michal Kolář  
• xkolar64  
• 187436

O Společnosti  
• Kontakty  
• Prodejna  
• O nás

Služby  
• Autorizovaný servis  
• Doprava  
• Možnosti platby  
• Záruka

Provozní doba  
• Po-Pá: 8:00-18:00  
• So-Ne: Zavřeno

Sledujte nás  
•   
•   
• 

## Příloha 6: Položka servisu

S

ko

Domů

iPhone

Mac

Odhlásit se

Košík (0)

Nová položka servisu: 1

X

Uživatel:

kolarm@gmail.com

Název položky:

iPhone11

Závada:

Rozbitý display

Stav:

Čeká na zaplacení

Souhlasím s opravou:

Ano

Ne

Cena v Kč (Ano):

11000

Cena v Kč (Ne):

600

Jméno na kartě

☐

Číslo karty

MM / RR CVC

Zaplatit a dokončit servis

Typ zařízení	Dle ceny	Dle typu	Dle velikosti	Reproduktor	Zadní kryt
iPhone 6s	2990Kč	1990Kč	990Kč	690Kč	890Kč
iPhone 7	3990Kč	1990Kč	1190Kč	690Kč	990Kč
iPhone 8	4990Kč	2090Kč	1990Kč	1190Kč	1990Kč
iPhone X	5990Kč	3000Kč	1990Kč	1690Kč	1890Kč
iPhone Xs	6990Kč	3990Kč	1990Kč	1690Kč	1890Kč
iPhone XR	6990Kč	3990Kč	1990Kč	1690Kč	1890Kč
iPhone 11	7990Kč	3990Kč	1990Kč	1690Kč	1890Kč
iPhone 11 Pro	8990Kč	3990Kč	1990Kč	1690Kč	1890Kč
Apple watch	5900Kč	2990Kč	-	-	-
Macbook Air 13'	10000Kč	4490Kč	-	4990Kč	-
Macbook Air 15'	11000Kč	5500Kč	-	5500Kč	-
Macbook PRO 13'	12000Kč	6000Kč	-	5500Kč	-
Macbook PRO 15'	13000Kč	6500Kč	-	6000Kč	-

Popis	Cena
Diagnostika	600Kč
Práce	800Kč
DPH 21%	Již započteno
Doprava	Zdarma